



Sun Java Enterprise System 术语表



Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

文件号码 819-4631
2007 年 3 月

版权所有 2007 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. 保留所有权利。

本文档及其相关产品的使用、复制、分发和反编译均受许可证限制。未经 Sun 及其许可方（如果有）的事先书面许可，不得以任何形式、任何手段复制本产品或文档的任何部分。第三方软件，包括字体技术，均已从 Sun 供应商处获得版权和使用许可。

本产品的某些部分可能是从 Berkeley BSD 系统衍生出来的，并获得了加利福尼亚大学的许可。UNIX 是 X/Open Company, Ltd. 在美国和其他国家/地区独家许可的注册商标。

Sun、Sun Microsystems、Sun 徽标、docs.sun.com、AnswerBook、AnswerBook2、Java 和 Solaris 是 Sun Microsystems, Inc. 在美国和其他国家/地区的商标或注册商标。所有的 SPARC 商标的使用均已获得许可，它们是 SPARC International, Inc. 在美国和其他国家/地区的商标或注册商标。标有 SPARC 商标的产品均基于由 Sun Microsystems, Inc. 开发的体系结构。

OPEN LOOK 和 SunTM 图形用户界面是 Sun Microsystems, Inc. 为其用户和许可证持有者开发的。Sun 感谢 Xerox 在研究和开发可视或图形用户界面的概念方面为计算机行业所做的开拓性贡献。Sun 已从 Xerox 获得了对 Xerox 图形用户界面的非独占性许可证，该许可证还适用于实现 OPEN LOOK GUI 和在其他方面遵守 Sun 书面许可协议的 Sun 许可证持有者。

美国政府权利 - 商业软件。政府用户应遵循 Sun Microsystems, Inc. 的标准许可协议，以及 FAR（Federal Acquisition Regulations，即“联邦政府采购法规”）的适用条款及其补充条款。

本文档按“原样”提供，对于所有明示或默示的条件、陈述和担保，包括对适销性、适用性或非侵权性的默示保证，均不承担任何责任，除非此免责声明的适用范围在法律上无效。

前言

本术语表确定 Java Enterprise System 的术语及定义。

另外，本术语表还将确定以下各项：

- 首字母缩写词
- 术语使用时的词性，例如，既当名词又当动词。所用到的词性缩写形式如下：
 - adj. - 形容词
 - n. - 名词
 - v. - 动词
- 以编号方式表示的、在不同产品或技术中的用法
- 交叉引用
- 同义词
- 对比术语
- 音标（适当时）

印刷约定

下表介绍了本书所采用的印刷约定。

表1 印刷约定

字体	含义	示例
AaBbCc123	命令、文件和目录的名称；计算机屏幕输出	编辑 .login 文件。 使用 <code>ls -a</code> 列出所有文件。 <code>machine_name% you have mail.</code>

表 1 印刷约定 (续)

字体	含义	示例
AaBbCc123	用户键入的内容，与计算机屏幕输出的显示不同	<code>machine_name%su</code> <code>Password:</code>
<i>AaBbCc123</i>	保留未译的新词或术语以及要强调的词。要使用实名或值替换的命令行变量。	要删除文件，请键入 <code>rm filename</code> 。
新词术语强调	新词或术语以及要强调的词。	高速缓存 是本地存储的副本。 切勿 保存文件。
《书名》	书名	阅读《用户指南》的第 6 章。

命令中的 Shell 提示符示例

下表显示了默认的系统提示符和超级用户提示符。

表 2 Shell 提示符

Shell	提示符
UNIX 和 Linux 系统上的 C shell	<code>machine_name%</code>
UNIX 和 Linux 系统上的 C shell 超级用户	<code>machine_name#</code>
UNIX 和 Linux 系统上的 Bourne shell 和 Korn shell	<code>\$</code>
UNIX 和 Linux 系统上的 Bourne shell 和 Korn shell 超级用户	<code>#</code>
Microsoft Windows 命令行	<code>C:\</code>

符号约定

下表说明本书中可能用到的一些符号。

表 3 符号约定

符号	说明	示例	含义
[]	包含可选参数和命令选项。	<code>ls [-l]</code>	-l 选项不是必需的。
{ }	包含所需命令选项的一组选择。	<code>-d {y n}</code>	-d 选项要求您使用 y 参数或 n 参数。

表 3 符号约定 (续)

符号	说明	示例	含义
<code>{ }</code>	表示变量引用。	<code>{com.sun.javaRoot}</code>	引用变量 <code>com.sun.javaRoot</code> 的值。
-	结合同时发生的多个击键。	Control-A	按 A 键的同时按 Control 键。
+	结合相继发生的多个击键。	Ctrl+A+N	按 Control 键后松开，然后按后续各键。
→	表示图形用户界面中的菜单项选择。	“文件” → “新建” → “模板”	从“文件”菜单中选择“新建”。从“新建”子菜单中选择“模板”。

文档、支持和培训

Sun Web 站点提供了有关以下附加资源的信息：

- 文档 (<http://www.sun.com/documentation/>)
- 支持 (<http://www.sun.com/support/>)
- 培训 (<http://www.sun.com/training/>)

搜索 Sun 产品文档

除了从 `docs.sun.com`SM Web 站点搜索 Sun 产品文档外，还可以使用搜索引擎进行搜索，方法是在搜索字段中键入以下语法：

```
search-term site:docs.sun.com
```

例如，要搜索 "broker"，请键入以下内容：

```
broker site:docs.sun.com
```

要将其他 Sun Web 站点包括到搜索范围内（例如，java.sun.com、www.sun.com 和 developers.sun.com），请在搜索字段中将 "docs.sun.com" 替换为 "sun.com"。

第三方 Web 站点引用

本文档所引用的第三方 URL 提供了其他相关信息。

注 - Sun 对本文档中提到的第三方 Web 站点的可用性不承担任何责任。对于此类站点或资源中的（或通过它们获得的）任何内容、广告、产品或其他资料，Sun 并不表示认可，也不承担任何责任。对于因使用或依靠此类站点或资源中的（或通过它们获得的）任何内容、产品或服务而造成的或连带产生的实际或名义损坏或损失，Sun 概不负责，也不承担任何责任。

Sun 欢迎您提出意见

Sun 致力于提高其文档的质量，并十分乐意收到您的意见和建议。要共享您的意见，请访问 <http://docs.sun.com>，然后单击 "Send Comments"（发送意见）。请在联机表单中提供完整的文档标题和文件号码。文件号码包含 7 位或 9 位数字，可在书的标题页或文档 URL 中找到该号码。例如，本书的文件号码是 819-4631。提出意见时您还需要在表格中输入文件的英文文件号码和标题。本文件的英文文件号码是 819-3875-14，文件标题为《Sun Java Enterprise System Glossary》。

A

abstract schema (抽象模式)	(n.)实体 bean 的部署描述符的一部分，定义 bean 的持久性字段和关系。参见 entity bean (实体 bean) 、 persistence (持久性) 。另参见 schema (模式) 。
abstract schema name (抽象模式名称)	(n.)EJB QL 查询中引用的逻辑名称。
access control (访问控制)	(1) (n.)通过对服务器的访问加以控制来保证服务器安全的方法。 (2) (n.)通过仅允许特定用户或程序集合与资源进行交互来实现完整性、保密性或可用性约束的方法。
access control entry (访问控制条目)	参见 ACE 。
access control instruction (访问控制指令)	参见 ACI 。
access control list (访问控制列表)	参见 ACL 。
access control rules (访问控制规则)	(n.)为给定的一组目录条目或目录属性指定用户权限的规则。
access domain (访问域)	(n.)一种域，用于限制从某一指定域内访问特定的 Messaging Server 操作。例如，访问域可用于限制可在何处收集帐户的电子邮件。
accessor (存取程序)	(n.)一种连接器层，可直接通过 LDAP 等协议与目录源相连。适用于 Windows 的身份认证服务拥有分别用于 Directory Server 、 Active Directory 和 Windows NT 的存取程序实现。在操作的日志消息中经常会提及到存取程序。
access rights (访问权限)	(n.)访问权限指定授予或拒绝的访问控制级别。访问权限与可在目录上执行的操作类型相关。可授予或拒绝下列权限：读、写、添加、删除、搜索、比较、自写、代理和全部。
account (帐户)	(n.)定义特定用户或用户组的信息。此信息包括用户名或组名、一个或多个有效电子邮件地址以及传送电子邮件的方式和位置。

account inactivation (帐户失效) (n.)禁用单个用户帐户或一组帐户，这样便可自动拒绝所有验证尝试。

ACE (access control entry, 访问控制条目) (1)(n.)访问控制列表中的单个信息项。也称为访问控制信息。

(2)(n.)一组具有层次结构的规则，Web 服务器使用它们来评估收到的访问请求。

(3)(n.)一个字符串，用于为日历、日历属性和日历组件（如事件和任务）提供访问控制。

ACI (access control instruction, 访问控制指令) (n.)一种指令，用于授予或拒绝对目录中条目的访问权限。

ACID (adj.)由 **transaction (事务)** 确保的四个属性的首字母缩略词：**atomicity (原子性)**、**consistency (一致性)**、**isolation (隔离性)** 和 **durability (持续性)**。

ACL (access control list, 访问控制列表) (1)(n.)用于对目录访问进行控制的机制。在 Directory Server 中，ACL 为目录条目中的一个 ACI 属性。

(2)(n.)ACE 集合。ACL 是定义哪些用户可访问您的服务器的一种机制。您可以定义特定于某个特殊文件或目录的 ACL 规则，准许或拒绝一个或多个用户和组的访问。

(3)(n.)一组 ACE 字符串，它们共同为日历、日历属性和日历组件（如事件和任务）提供访问控制。

(4)(n.)与目录相关联的一组数据，它们定义用户、组或用户及组所具有的访问该目录的权限。ACL 由一个或多个 ACE 字符串组成。

account federation (帐户联合) 参见 **identity federation (身份联合)**。

accumulated patch (累积修补程序) (n.)一个修补程序，该程序结合了先前的一个或多个修补程序中的补丁、相同修补程序的任意先前版本中的补丁和将要发布的当前补丁集。

activation (活化) (n.)将企业 bean 的状态从辅助存储器转移到内存的过程。另参见 **passivation (钝化)**。

active boot environment (活动引导环境) (n.)当前已开启并运行的环境。

active node (活动节点) (n.)包含会话数据的 HADB 节点。如果活动节点出现故障，备用节点会从镜像节点复制数据并转为活动状态。另参见 **HADB node (HADB 节点)**、**spare node (备用节点)**、**mirror node (镜像节点)** 及 **data redundancy unit (数据冗余单元)**。

address (地址) (n.)电子邮件消息中用于确定消息发送方式和位置的信息。消息头和消息信封中都包含地址。信封地址确定如何路由和传送消息。头地址仅用于显示。

address handling (地址处理) (n.)由 MTA 执行的操作，其目的是检测寻址过程中的错误、必要时重写地址以及将地址与收件人进行匹配。

addressing protocol (寻址协议) (n.)使电子邮件可以传送的寻址规则。RFC 822 是在 Internet 上应用最广的协议，也是 Messaging Server 支持的协议。其他协议还有 X.400 和 UUCP。

address token (地址令牌)	(n.) 重写规则模式的地址元素。
admin console (管理控制台)	(n.) 一组基于浏览器的表单，用于配置、管理、监视、维护和故障排除 Java™ Enterprise System 服务器及其组件。 (n.) Portal Server 6.0 对于管理员的 Directory Server Access Management Edition GUI 界面。
administered object (受管对象)	(n.) 由管理员创建的预配置 Java Enterprise System 对象 (连接工厂或目的地)，由一个或多个 JMS 客户端使用。 使用受管对象可将 Java 消息服务 (Java Message Service, JMS) 客户端与提供者的专有特征隔离开来。这些对象由管理员置入一个 Java Naming and Directory Interface™ (JNDI) 名称空间中，并由 JMS 客户端使用 JNDI 查找进行访问。
administration console (管理控制台)	参见 admin console (管理控制台) 。
administration domain (管理域)	参见 domain (域) 。
administration interface (管理界面)	参见 admin console (管理控制台) 。
administration node (管理节点)	(n.) 可与远程管理服务器通信的 Web Server 节点。 cluster (群集) 或 server farm (服务器群) 中的每个节点上都运行有管理服务器或管理节点。在这些节点中，将某个节点配置为主服务器，称为管理服务器；将其他节点配置为从属服务器，称为管理节点。
administration privileges (管理权限)	(n.) 定义用户管理角色的一组权限。
administration server (管理服务器)	(n.) 提供 Java Enterprise System 组件产品管理功能的特殊服务器。
administration server administrator (管理服务器管理员)	(n.) 即使在没有 Java Enterprise System Directory Server 连接的情况下亦具有启动或停止服务器管理权限的用户。管理服务器管理员承担本地服务器组中所有服务器的有限服务器任务 (通常只是“重新启动服务器”和“停止服务器”)。安装管理服务器时，此管理员的条目自动在本地创建。此管理员不是用户目录中的用户。
administrative domain (管理域)	参见 domain (域) 。
administrator (管理员)	(n.) 具有定义的一组管理权限的用户。另参见 configuration administrator (配置管理员) 、 Directory Manager (目录管理员) 、 administration server administrator (管理服务器管理员) 、 server administrator (服务器管理员) 、 family group administrator (系列组管理员) 、 mail list owner (邮件列表所有者) 。
admpw	(n.) Sun Enterprise™ Administrator Server 超级用户的用户名和密码文件。
adoption scenario (应用情况)	部署 Java Enterprise System 软件的总体原因，描述您开始使用的软件系统以及要达到的目标。有四种基本的 Java Enterprise System 应用情况：新系统、替换、扩充和升级。

- affiliation (联合提供者)** (n.)联合提供者是一组提供者, 当他们组成联合提供者时, 不考虑他们的特殊验证域。联合提供者由联合提供者所有者形成并维护。联合提供者文档描述由各提供者的 providerID 共同标识的一组提供者。联合提供者的成员可以通过成员身份 (利用其“联合提供者 ID”) 调用服务, 也可以单独 (利用其“提供者 ID”) 调用服务。
- agent (代理)** (1) (n.)在网络设备 (如路由器、主机或 X 终端) 中运行网络管理软件的软件。另参见 [intelligent agent \(智能代理\)](#)、[node agent \(节点代理\)](#)。
(2) (n.)在适用于 Windows 的身份认证服务中, 代理是一种连接器组件, 可与 Message Queue 进行交互并在其 Directory Server 名称与 Windows 名称之间转换属性。在操作的日志消息中经常会提及到代理。
- alarm event (报警事件)** (n.)由 Calendar Server ENS 生成并发送的事件。发生报警事件时, 会向特定收件人发送一个消息提示。
- alert (警报)** (n.)用户在弹出窗口即时收到的紧急消息。只要使用 "Show message status" (显示消息状态) 选项, 发件人便可在关闭或单击警报时知道谁收到了消息并得到消息已被阅读的通知。如果该警报消息要求响应, 可右键单击警报以调用一个上下文菜单, 其中有 (与发件人聊天) 选项。
- alias file (别名文件)** (n.)用于设置未在目录中设置的别名 (如邮寄主管别名) 的文件。
- aliasing (别名设置)** (n.)在 Java Enterprise System Portal Server 搜索引擎中用一项替代另一项, 该搜索引擎在从具有不同模式的其他搜索引擎导入资源描述时, 将会使用别名设置。
- All IDs threshold (全部 ID 阈值)** (n.)全局应用于 Java Enterprise System Directory Server 所管理的每个索引的大小限制。当某一条目 ID 列表达到此限制值时, 服务器用一个“全部 ID”令牌替换该条目 ID 列表。
- All IDs token (全部 ID 令牌)** (n.)致使服务器假定所有目录条目都与索引键匹配的一种机制。实际上, “全部 ID”令牌会使 Java Enterprise System Directory Server 执行无索引搜索来匹配索引键。
- allowed attributes (允许的属性)** (n.)可存在于使用特定对象类的条目中的可选属性。另参见 [attribute \(属性\)](#)、[required attributes \(必需属性\)](#)。
- Allow filter (允许过滤器)** (n.)Java Enterprise System Messaging Server 访问控制规则, 用于确定允许访问一个或多个 POP、IMAP 或 HTTP 服务的客户机。另参见 [deny filter \(拒绝过滤器\)](#)。
- alternate address (备用地址)** (n.)某帐户的辅助地址, 通常是主地址的变型。在某些情况下, 一个帐户有多个地址会更方便一些。
- alternate root (备用根)** (n.)客户机上 root 文件系统中安装某一软件包的位置。备用根一般是使用 pkgadd -R 提供的。
- AML** (abstract markup language, 抽象标记语言) (n.)独立于特定供应商或型号的移动设备标记语言。
- anonymous access (匿名访问)** (1) (n.)没有进行 [authentication \(验证\)](#) 就访问 [resource \(资源\)](#)。
(2) (n.)此种访问经授予后允许任何人不提供凭证即可访问目录信息, 无论绑定情况如何。

API	(application programming interface, 应用编程接口) (1) (n.) 一组指令, 计算机程序可使用这些指令与专用于解释该 API 的其他软件或硬件进行通信。 (2) (n.) 一组调用约定或指令, 定义程序如何调用现有软件包中的服务。
APOP	(authenticated post office protocol, 验证邮局协议) 与 POP 类似, 但 APOP 不使用明文密码进行验证, 而是将密码编码与盘问字符串结合起来使用。
applet container (applet 容器)	(n.) 一种支持 applet 编程模型的 container (容器) 。
application assembler (应用程序组合者)	(n.) 将 J2EE™ component (组件) 和 module (模块) 组合到可以部署的应用程序单元的人员。
application client (应用程序客户端)	(n.) 在其自身的 Java 虚拟机上执行的第一层 J2EE 客户机组件。应用程序客户端可访问某些 J2EE 平台 API。
application client container (应用程序客户端容器)	(n.) 一种支持应用程序客户端组件的容器。参见 container (容器) 。
application client module (应用程序客户端模块)	(n.) 由一个或多个类以及一个应用程序客户端部署描述符组成的软件单元。
application component (应用程序组件)	参见 component (组件) 。
application component provider (应用程序组件提供者)	(n.) 提供 Java 类来实现组件方法、JSP 页定义和所有必需的部署描述符的供应商。
application configuration resource file (应用程序配置资源文件)	(n.) 一种 XML 文件, 用于为 JavaServer Faces 应用程序配置资源、为应用程序定义导航规则以及将转换器、验证器、侦听器、呈现器和组件注册到应用程序中。
Application Server	(n.) 包含在 Sun Java Enterprise System 中的应用服务器产品。
application server (应用服务器)	(n.) 运行业务应用程序的一种软件平台。应用服务器通常为应用程序提供高级别服务, 如组件生命周期、位置以及分发和事务资源访问。
application service (应用服务)	(n.) 组件或组件集合体, 代表多个客户机执行业务逻辑, 因而必须是一个多线程进程。应用程序服务也可以是封装为 Web 服务或独立的内容服务器的组件或组件集合体。
application tier (应用层)	(n.) 对 J2EE 应用程序在概念上的划分: 客户层 : 用户界面。最终用户与客户机软件 (如 Web 浏览器) 交互, 以使用应用程序。 服务器层 : 组成应用程序的业务逻辑和表示逻辑, 在应用程序的组件中定义。 数据层 : 使您的应用程序能够与数据源交互的数据访问逻辑。

- approximate index (近似索引)** (n.)一种索引, 用以在目录信息树中进行有效的近似或“音近”搜索。
- architecture (体系结构)** 一种设计, 展示了分布式应用程序 (或其他某个软件系统) 的逻辑和物理构件及其相互关系。就 **distributed enterprise application (分布式企业应用程序)** 而言, 体系结构设计通常同时包括应用程序的 **logical architecture (逻辑体系结构)** 和 **deployment architecture (部署体系结构)**。
- archiving (归档)** (n.)保存和恢复对象状态的过程。
- A record (A 记录)** (n.)一种 DNS 记录, 包含主机名及其关联的 IP 地址。A 记录由 Internet 上的消息传送服务器用于为电子邮件选择路由。另参见 **domain name system (域名系统)**、**MX record (MX 记录)**。
- asant** (n.)一种基于 Apache Ant 的生成工具, 可使用 Java 对其进行扩展。配置文件基于 XML, 并调用在其中执行各类任务的目标树。另参见 **build file (生成文件)**。
- assembly (装配)** (n.)将某一应用程序的分散组件组合到可以部署的单个单元中的过程。另参见 **deployment (部署)**。
- asynchronous communication (异步通信)** (n.)一种通信模式, 在该模式下, 消息的发送者不需要等待发送方法返回即可继续其他工作。
- attribute (属性)**
- (1) (n.)请求对象中可由 servlet 设置的名称值对。也可以是 DTD 文件中预定义的用于修改 XML 文件中某一元素的名称值对。与 **property (属性)** 不同。另参见 **parameter (参数)**。属性更通常指一个元数据单位。
 - (2) (n.)用于存放条目描述信息的名称值对。属性具有类型 (名称) 和一组值。属性类型还为可以存储为该属性值的信息种类进行了语法规定。
 - (3) (n.)定义 Java Enterprise System Directory Server Access Management Edition 服务向组织提供的参数。组成 Java Enterprise System Directory Server Access Management Edition 服务的属性可以归为以下任一类别: 动态、策略、用户、组织或全局。使用这些类型对每种服务中的属性进行细分, 可以使服务模式管理更一致并可简化服务参数的管理。
 - (4) (n.)Application Server 中作为内置服务器配置的一部分的名称值对。与 **property (属性)** 不同。
- attribute provider (属性提供者)** (n.)属性提供者是托管属性数据的一种 Web 服务。
- attribute list (属性列表)** 参见 **optional attribute list (可选属性列表)** 和 **required attribute list (必需属性列表)**。
- auditing (审计)** (n.)记录重要事件以备后续检查 (通常在出错或安全性被破坏的情况下进行) 的方法。
- AUTH** (n.)一个 SMTP 命令, 借此, SMTP 客户机能够指定对服务器的验证方法、执行验证协议交换, 以及在必要时对后续协议交互的安全层进行协商。

- authenticating Directory Server (验证 Directory Server)** (n.)在 PTA 中, 验证 Java Enterprise System Directory Server 包含请求客户机的验证凭证。启用 PTA 的用户目录将绑定请求传递给验证目录, 该目录检验请求方客户机的绑定凭证。
- authentication (验证)** (1)(n.)验证计算机系统中用户、设备或其他实体的身份的过程, 该过程通常是允许访问系统中资源的先决条件。Java servlet 规范需要三类验证 (基本验证、基于窗体的验证和相互验证) 并支持摘要验证。在专用或公共计算机网络 (包括 Internet) 中, 验证通常使用登录密码完成。知道密码的用户即被视为可信用户。另参见 [basic authentication \(基本验证\)](#)、[form-based authentication \(基于窗体的验证\)](#)、[mutual authentication \(相互验证\)](#) 和 [digest authentication \(摘要验证\)](#)。
- (2)(n.)向 Java Enterprise System Directory Server 证明客户机用户身份的过程。用户必须提供一个绑定 DN 和相应的密码或证书, 才能被授权访问目录。Java Enterprise System Directory Server 允许用户在经目录管理员授予权限后执行功能或访问文件和目录。另参见 [server authentication \(服务器验证\)](#)。
- authentication certificate (验证证书)** (n.)从服务器发送到客户机或从客户机发送到服务器以对另一方进行检验或验证的数字文件。证书确保其持有者 (客户机或服务器) 的真实性。证书不可转让。
- authentication domain (验证域)** (n.)一组服务提供者, 其中至少有一个同意使用“自由联盟计划”(Liberty Alliance Project, LAP) 交换用户验证信息的身份认证提供者。一旦建立了 [circle of trust \(信任环\)](#), 即在所有提供者之间启用了单点登录验证。也称为信任环。
- authorization (授权)** (n.)确定主体是否可使用服务、主体可访问哪些对象以及每个对象可使用的访问类型的过程。授权取决于通过验证与请求相关的主体是否处于给定的安全角色中。安全角色是组合应用程序的人员所定义的用户逻辑组。部署者将安全角色映射到安全身份。安全身份可能是操作环境中的主体或组。
- authorization constraint (授权约束)** (n.)一种授权规则, 用于确定允许其访问 Web 资源集合的人员。
- autoreply option file (自动回复选项文件)** (n.)用于设置电子邮件自动回复选项 (如休假通知) 的文件。
- AutoReply utility (AutoReply 实用程序)** (n.)一个实用程序, 可对发送给激活了 AutoReply 功能的帐户的消息自动做出响应。Java Enterprise System Messaging Server 中的每个帐户都可配置为自动回复收到的消息。
- availability service (可用性服务)** (n.)Application Server 的一种功能, 用于在服务器实例、Web 容器、EJB 容器上启用高可用性, 也用于为 RMI/IIOP 请求启用高可用性。

B

B2B	(adj.)Business-to-business (企业到企业)。
backbone (主干)	(n.)分布式系统的主要连接机制。与主干上的一个中间系统连接的所有系统都彼此连接。主干不会妨碍您出于成本、性能或安全原因而将系统设置为绕过主干。
back-end server (后端服务器)	(n.)在 Java Enterprise System Messaging Server 上下文中, 指的是只具备存储和检索电子邮件消息功能的电子邮件服务器。也称为消息存储服务器。
backing bean (支持 bean)	(n.)一种 JavaBeans 组件, 对应于包括 JavaServer Faces 组件的 JSP 页。支持 bean 定义页面上组件的属性以及执行组件处理过程的方法。该处理过程包括与导航相关的事件处理、验证以及处理。
backout (回退)	(n.)将软件更改(如修补程序)删除, 从而使系统返回先前状态。
back up (备份)	(v.)将消息存储中的文件夹内容复制到备份设备。另参见 restore (恢复) 。
backup store (备份存储)	(n.)数据的系统信息库, 通常为文件系统或数据库。可由后台线程(sweeper 线程)监视备份存储以便移除不需要的条目。
banner (标题)	(n.)客户机首次连接到某一服务(如 IMAP)时该服务显示的文本字符串。
base DN (基 DN)	(base distinguished name, 基标识名) (n.)DIT 中的一个条目。对于由基 DN 标识的条目、直接从属于该基 DN 或该条目的条目以及 DIT 中该基 DN 之下的所有条目, 都可执行搜索操作。
basic authentication (基本验证)	(n.)Web 服务器通过由 Web 应用程序内置验证机制获取的用户名和密码来对实体进行验证的验证机制。
bean-managed persistence (bean 管理持久性)	(n.)由 entity bean (实体 bean) 来管理实体 bean 的变量与资源管理器之间的数据传送的机制。数据访问逻辑通常由开发者使用 Java™ 数据库连接 (Java Database Connectivity, JDBC™) 软件或其他数据访问技术提供。另参见 container-managed persistence (容器管理持久性) 。
bean-managed transaction (bean 管理事务)	(n.)企业 bean 的事务划分由开发者通过程序控制。另参见 container-managed transaction (容器管理事务) 。

Berkeley DB	(Berkeley database, Berkeley 数据库) (n.)一种事务数据库存储, 专用于高并发性读写工作负载以及需要事务和可恢复性的应用程序。Java Enterprise System Messaging Server 将 Berkeley 数据库用于许多用途。
binary entity (二进制实体)	(n.)参见 unparsed entity (未解析实体) 。
bind DN (绑定 DN)	(bind distinguished name, 绑定标识名) (n.)绑定请求中用于对 Java Enterprise System Directory Server 进行验证的标识名。
binding (绑定)	(1) (v.)对于 XML 文件, 生成处理明确定义的 XML 数据部分所需的代码。 (2) (v.)对于 JavaServer Faces 技术, 将 UI 组件连接到后端数据源 (如支持 bean 属性)。
bind rule (绑定规则)	(n.)在访问控制环境中, 绑定规则指定特定用户或客户机要访问目录信息所必须具备的凭证和满足的条件。
BLOB	(binary large object, 二进制大对象) (n.)用于存储和检索复杂对象字段的一种数据类型。BLOB 是二进制或可序列化对象 (如图片), 转换后为大字节数组, 随后序列化为容器管理持久性字段。
BMP	参见 bean-managed persistence (bean 管理持久性) 。
BMT	参见 bean-managed transaction (bean 管理事务) 。
body (正文)	(n.)电子邮件消息的一部分。尽管消息头和信封必须遵循标准格式, 但消息正文的内容可由发件人决定。正文可包含文本、图形或多媒体。结构化正文遵循 MIME 标准。
BPEL	(business process execution language, 业务流程执行语言) (n.)扩展交互式流程的 web service (Web 服务) 的业务流程语言。以 XML 表示法表达 BPEL 流程。
broker (代理)	(n.)负责管理 Java 消息服务 (Java Message Service, JMS) API 消息路由选择、传送、持久性、安全性和日志记录的 Message Queue 实体。代理提供了一个接口, 管理员可使用该接口监视并调优性能和资源的使用。
browsing (浏览)	(n.)在 Java Enterprise System Portal Server 内, 指在某一“搜索”数据库中查找按类别划分的资源。
browsing index (浏览索引)	参见 virtual list view index (虚拟列表视图索引) 。
build file (生成文件)	(n.)包含一个或多个 asant 目标的 XML 文件。目标是一组要执行的 Apache Ant 任务。可在启动 asant 时选择要执行的目标。如果没有给定目标, 则会执行项目的默认目标。另参见 asant 。
building module (构建模块)	(n.)不依赖或部分依赖共享服务的硬件或软件结构。是一种可提供最佳性能和水平可伸缩性的特定配置。
business logic (业务逻辑)	(n.)实现某一应用程序 (而非数据集成或显示逻辑) 的基本功能的代码。在 EJB technology (EJB 技术) 中, 通过企业 bean 的方法来实现该逻辑。
business method (业务方法)	(n.)实现应用程序业务逻辑或规则的企业 bean 方法。

business service (业务服务) **application component (应用程序组件)** 或组件的集合体，代表多个客户端执行业务逻辑（因而是一个多线程进程）。业务服务也可以是作为 **web service (Web 服务)** 封装起来的分布式组件的集合体，还可以是独立的 **server (服务器)**。

C

- CA** (1) (certificate authority, 证书授权机构) (n.)参见 [certificate authority](#) (证书授权机构)。
(2) (connector architecture, 连接器架构) (n.)参见 [connector architecture](#) (连接器架构)。
- cache (高速缓存)** (n.)本地存储的原始数据的副本。请求使用缓存的数据时, 不必再从远程服务器检索。
- Cache Control Directive (高速缓存控制指令)** (n.)Java Enterprise System Application Server 控制代理服务器缓存哪些信息的一种方法。您可使用缓存控制指令覆盖代理的默认缓存, 避免缓存敏感信息, 从而防止以后被检索。要使这些指令起作用, 代理服务器必须与 HTTP 1.1 兼容。
- cached rowset (缓存行集)** (n.)一种对象, 允许您从某数据源检索数据, 然后在检查和修改数据时与该数据源分离。缓存行集既跟踪检索的原始数据, 也跟踪应用程序对原始数据所做的任何更改。如果应用程序尝试更新原始数据源, 则此行集被重新连接到该数据源, 而且只有那些发生更改的行被合并回数据库中。
- calendar access protocol (日历访问协议)** 参见 [CAP](#)。
- Calendar Express** (n.)一种基于 Web 的日历客户机程序, 为最终用户提供对 Calendar Server 的访问权。
- calendar group (日历组)** (n.)若干日历的集合, 可帮助用户管理多个日历。
- calendar ID (日历 ID)** (n.)与 Java Enterprise System Calendar Server 数据库中的某一日历关联的唯一标识符。也称为 calid。
- calendar lookup database (日历查找数据库)** 参见 [CLD](#)。
- Calendar Server application programming interface (Calendar Server 应用编程接口)** 参见 [CSAPI](#)。

- calendar user agent (日历用户代理)** 参见 [CUA](#)。
- callable statement (可调用语句)** (n.)一个类，对于支持从存储过程返回结果集的数据库，该类可封装数据库的过程和函数调用。
- callback method (回调方法)** (n.)一种组件方法，在组件的生命周期过程中，容器调用此方法来向其通知重要事件。
- caller (调用方)** (n.)等同于调用方主体。
- caller principal (调用方主体)** (n.)标识企业 bean 方法的调用者的主体。
- CAP** (calendar access protocol, 日历访问协议) (n.)一种标准的 Internet 日历协议，基于 Internet 工程任务组 (Internet Engineering Task Force, IETF) 确定的要求。
- capability (能力)** (n.)为客户机提供的字符串，它定义某一给定 IMAP 服务中的可用功能。
- cascading deletion (级联删除)** (n.)触发其他删除的删除。可为具有 [container-managed persistence \(容器管理持久性 \)](#) 的 [entity bean \(实体 bean \)](#) 指定级联删除。
- cascading replication (级联复制)** (n.)在级联复制方案中，一台服务器（通常称为集线器供应者）既是特定副本的使用方同时又是供应者。该服务器保留一份只读副本并维护更改日志。它从持有原版数据的供应者服务器接收更新内容，然后再将这些更新提供给使用者。
- catalog (目录)** 参见 [index \(索引 \)](#)。
- cataloging (编目)** 参见 [indexing \(建立索引 \)](#)。
- category (类别)** (n.)“搜索”数据库内资源的逻辑分组。概括而言，一组类别有时也称为“分类系统”。
- CCPP** (composite capability and preference profiles, 功能及首选项综合配置文件) (n.)对于 Portal Server Mobile Access 软件而言，它是用于“用户代理配置文件”的规范及用于客户机检测的预配置数据。CCPP 规范描述了设备的功能和用户首选项。
- CDATA** (n.)一种预定义的针对字符数据的 XML 标记，表示“不解释这些字符”（与解析的字符数据 (PCDATA) 相对，此类数据中应用常规的 XML 语法规则）。CDATA 部分通常用于显示 XML 语法示例。另参见 [PCDATA](#)。
- central logger (中心记录器)** (n.)管理所有中心日志的 Core 组件，这些中心日志聚合了每个连接器的审计和错误日志。管理员可通过监视这些日志来监视整个适用于 Windows 的身份认证服务 安装的运行状况。可直接查看中心日志，也可从适用于 Windows 的身份认证服务 控制台查看中心日志。默认情况下，如果在机器的 `<install-root>/logs/central/` 子目录下安装有 Core 组件，则该机器中存在中心日志。
- certificate (证书)** (1) (n.)用于标识 Instant Messaging Server 的电子文档，与公共密钥关联在一起。Java Enterprise System Instant Messaging Server 支持 Instant Messaging 服务器之间交换证书。证书交换对于单个用户来说是透明的。
- (2) (n.)一种数字数据，指定个人、公司或其他实体的名称并证明证书内所含公共密钥属于该实体。客户机和服务器都可拥有证书。

- (3) (n.)证书将用户或 CA 的公共密钥与该用户或 CA 的身份 (通常为标识名) 紧密关联。证书由 CA 以数字化方式签署, 并可在建立 SSL 连接期间验证, 以获得连接另一端的公共密钥。X.509 证书存储在 `caCertificate;binary` 或 `userCertificate;binary` 属性中的目录中。
- certificate authority (证书授权机构)** (1) (n.)内部或第三方受信组织, 签发用于加密事务的公钥证书, 并为载送方提供标识。
- (2) (n.)网络中签发及管理安全凭证和用于消息加密和解密的公共密钥的授权机构。作为 PKI 的一部分, CA 与注册授权机构核对数字证书申请者提供的信息, 以进行检验。如果申请者的信息通过了注册授权机构的检验, 则 CA 可以签发证书。另参见 [PKI](#)。
- certificate-based authentication (基于证书的验证)** (n.)根据客户机提交的数字证书确认用户的身份。另参见 [password authentication \(密码验证 \)](#)。
- certificate database (证书数据库)** (n.)包含服务器的数字证书或证书的文件。也叫做证书文件。
- certificate name (证书名称)** (n.)标识证书及其所有者的名称。
- certificate revocation list (证书撤销列表)** 参见 [CRL](#)。
- CGI** ([common gateway interface](#), 公共网关接口) (n.)外部程序与 HTTP 服务器通信所用的接口。使用 CGI 的程序被称为 CGI 程序或 CGI 脚本。CGI 程序处理的表单或解析的输出通常是服务器无法处理或解析的。
- cHTML** (n.)适用于移动设备的简化版 HTML。
- change log (更改日志)** (n.)更改日志是对副本所作修改的记录。然后供应者服务器将在存储于使用者服务器上的副本上回放这些修改, 或者在多主副本复制情况下, 在其他主副本上回放这些修改。注意, 它与追溯更改日志不同, 后者不用于复制。
- channel (通道或频道)** (1) (n.)处理消息的基本 MTA 组件。通道代表与另一计算机系统或一组系统的连接。每个通道由一个或多个通道程序和一个外出的消息队列组成, 该队列用于存储将要发送给与该通道相关联的一个或多个系统的消息。另参见 [channel block \(通道块 \)](#)、[channel host table \(通道主机表 \)](#)、[channel program \(通道程序 \)](#)。
- (2) (n.)在 Java Enterprise System Portal Server Desktop 中, 频道由提供者和配置组成。频道生成的内容可包括标记片段、框架集、HTML 页面等。频道内容常常与其他频道内容汇集, 形成“Portal 桌面”。
- channel block (通道块)** (n.)单一通道定义。另参见 [channel host table \(通道主机表 \)](#)。
- channel host table (通道主机表)** (n.)通道定义的集合。另参见 [channel block \(通道块 \)](#)。

- channel program (通道程序)** (n.)通道的一部分，负责将消息发送到远程系统，并在发送后从队列中删除消息，以及从远程系统接收消息并将它们放置到适当的通道队列中。另参见 [master channel program \(主通道程序\)](#)、[slave channel program \(从通道程序\)](#)。
- character type (字符类型)** (n.)用于区分字母型字符与数值或其他类型字符的属性，以及大小写字母映射。
- chat (聊天)** (n.)Instant Messaging 的即时消息传送版本。“聊天”是一项实时会话功能。“聊天”会话可以在根据需要随时创建的聊天室或预先建好的会议室内进行。
- checkpoint (检查点)** (n.)有状态会话 bean 的生命周期中的一个预定义点，如果 Application Server 实例失败，则会在该点将 bean 的状态保存到持久性存储中。
- child (子)** (1) (n.)其他类别的子类别。另参见 [category \(类别\)](#)。
(2) (n.)XML 文件中的元素，包含在另一个称为“父”的元素内。另参见 [parent \(父\)](#)。
- chroot** (n.)可以创建的附加根目录，用于将服务器限制在特定目录内。可以使用此功能来保护未受保护的服务器。
- cipher (加密器)** (n.)加密器是用于加密或解密的加密算法 (数学函数)。
- ciphertext (密文)** (n.)没有正确密钥解密信息时任何人都无法读取的加密信息。
- circle of trust (信任环)** (n.)参见 [authentication domain \(验证域\)](#)。
- CKL** (compromised key list, 已泄密密钥列表) (n.)拥有已泄密密钥的用户的密钥信息列表。CA 也提供此列表。术语“已泄密密钥列表”的附加定义。另参见 [CRL](#)。
- classic CoS (传统 CoS)** (n.)用其 DN 和一个目标条目属性的值标识模板条目。
- classification rules (分类规则)** (n.)用于将资源指定给某一类别或几个类别的一组规则。
- class loader (类加载器)** (n.)基于 Java™ 技术的组件，负责按照特定规则加载 Java 类。
- class of service (服务类)** 参见 [CoS](#)。
- CLD** (Calendar Lookup Database, 日历查找数据库) (n.)一个插件，当日历数据库分布于两个或多个后端服务器时，该插件用于确定日历的物理位置。Calendar Server 提供 LDAP CLD 插件和算法 CLD 插件。
- cleartext (明文)** (n.)未加密文本。
- client-certificate authentication (客户机证书验证)** (n.)通过 SSL 使用 HTTP 的验证机制，在该机制中，服务器和 (可选) 客户机通过公钥证书相互进行验证，该公钥证书需符合 X.509 公钥基础结构所定义的标准。另参见 [authentication \(验证\)](#)、[certificate authority \(证书授权机构\)](#)。
- client contract (客户机订约)** (n.)一种订约，它确定客户机和 EJB™ 容器之间的通信规则，为使用企业 bean 的应用程序建立统一开发模型，并通过标准化与客户机之间的关系来确保 bean 的更大再使用率。

client conditional properties (客户机条件属性)	(n.)Portal Server Mobile Access 客户机类型的属性，它允许管理员为给定客户机的频道或容器频道指定属性。
client database (客户机数据库)	(n.)对于 Portal Server Mobile Access 而言，它是由内部和外部库组成的数据库。内部库包含所有默认移动设备数据定义。外部库包含自定义的客户机数据定义，这些定义会覆盖内部库中的定义。
client detection (客户机检测)	(n.)用于确定访问门户的每个移动设备的功能和特性的 Access Manager 进程。
Client Editor (客户机编辑器)	(n.)允许用户创建客户机类型并管理客户机属性的 Access Manager 界面。Client Editor 界面可从 Access Manager 控制台访问。
client identifier (客户机标识符)	(n.)将连接及其对象与某状态关联在一起的标识符，它由 Java Enterprise System 消息服务器代表客户机来维护。
Client Manager (客户机管理器)	(n.)可从控制台访问的 Access Manager 界面，利用它可管理客户机类型和属性。
client profile (客户机配置文件)	(n.)标识每个客户机的 Access Manager 配置文件。
*client runtime (客户机运行时环境)	参见 Java Enterprise System 客户机运行时环境。
client-server model (客户机/服务器模型)	(n.)一种计算模型，其中联网计算机向其他客户机计算机提供特定的服务。例如，DNS 的名称服务器和名称解析器范例，以及文件服务器/文件客户机之间的关系（如 NFS 和无硬盘主机）。
*client type (客户机类型)	(n.)Access Manager 客户机数据库内的条目。
clientType	(n.)一种属性，它指的是为 Access Manager 客户机数据提供唯一索引的名称。
cluster (群集)	(1)(n.)由高速网络连接在一起的一组服务器、代理或节点，它们像一台服务器、一个代理或一个节点一样协同工作。如果群集内的一台服务器、一个代理或一个节点出现故障，则其服务可故障转移至另一正常运行的服务器、代理或节点。另参见 broker (代理)、 failover (故障转移)、 node (节点)、 server (服务器)。
CMP	参见 container-managed persistence (容器管理持久性)。
CMR	参见 container-managed relationship (容器管理关系)。
CMT	参见 container-managed transaction (容器管理事务)。
cn	参见 common name attribute (通用名称属性)。
CNAME record (CNAME 记录)	(n.)一种将域名别名映射到域名的 DNS 记录。
collation order (对照顺序)	(n.)一种与语言及文化相关的信息，说明如何排序给定语言的字符。此信息可能包括字母表中字母的顺序或如何比较有重音字母与无重音字母。

collection (集合)	(n.)包含文档信息的数据库，如词汇表或文件属性。搜索功能将使用集合来检索符合指定搜索条件的文档。
collocation (并置)	(n.)一种属性，说明位于同一节点上。此概念在群集配置期间使用，以改善性能。
colocate (协同定位)	(v.)将某组件和与其相关组件置于同一内存空间处，以避免远程过程调用从而改善性能。
column (列)	(n.)数据库表内的字段。
comm_dssetup.pl	(n.)一种 Directory Server 准备工具，它可以使现有的 Directory Server 做好供 Messaging Server 使用的准备。
comment (注释)	(n.)XML 文本中除解析程序专门要求识别外均可忽略的文本。
comment character (注释字符)	(n.)位于行首、将该行转变为非可执行注释的字符。
commit (提交)	(1) (v.)通过向数据库或其他资源发送所需命令来完成某项事务。另参见 rollback (回滚) 、 transaction (事务) 。 (2) (n.)事务中的某个时刻，在该时刻，对事务中所有资源进行的更新都会设为永久性更新。
common domain (通用域)	(n.)在有多个身份认证提供者的 circle of trust (信任环) 中，服务提供者需要一个能够确定主体将使用哪一身份认证提供者的方法。由于此功能必须跨任一数量的域名系统 (domain name system, DNS) 域工作，所以 Liberty 采用了为环内所有身份和服务提供者创建一个通用域的方法。此预定域称为通用域。在通用域内，当主体通过服务提供者的验证后，身份认证提供者将编写一个用于存储主体的身份认证提供者的通用域 cookie。然后，当主体尝试访问环内其他服务提供者时，该服务提供者将读取此通用域 cookie，请求便可转发给正确的身份认证提供者。
common log file format (通用日志文件格式)	(n.)服务器将信息输入访问日志时使用的格式。所有主要服务器（包括 Web Server）之间的格式均相同。
common name attribute (通用名称属性)	(n.)标识由 LDAP 目录内条目定义的人员或对象的 cn 属性。
Communication Services (通信服务)	(n.)一种综合消息传送解决方案，可将集成电子邮件、日历、即时消息传送以及在线状况信息传送给企业客户。“通信服务”核心解决方案包括 Messaging Server、Calendar Server 和 Instant Messaging Server。
Communications Express	(n.)一种软件，它提供了基于 Web 的集成式通信和协作客户机，用以满足企业用户访问电子邮件、日历和通讯录信息的需要。
Compass	(n.)一种为 Portal Server 3.0 提供搜索功能的搜索引擎服务。该搜索引擎已集成到 Portal Server 6.0 的内核中。参见 Search Engine (搜索引擎) 。
Compass Server	(n.)用于方便用户访问网络资源的服务器技术，通常与 Portal Server 3.0 一同使用。Portal Server 6.0 包含一个紧密集成的搜索引擎，该引擎具有 Compass Server 对 Portal Server 3.0 提供的功能。
component (组件)	(1) (n.)包含在 Java Enterprise System 内的 system component (系统组件) 之一。

	(2)(n.)一个软件逻辑单元，分布式应用程序就是由它建立出来的。应用程序组件是自定义开发的，通常符合分布式组件模型（如 CORBA 和 J2EE 平台），并执行某些特定的计算功能。这些组件单独或组合起来提供 business service （业务服务），并且可以封装为 web service （Web 服务）。
	(3)(n.)参见 J2EE component （J2EE 组件）。
component contract （组件约定）	(n.)J2EE 组件与其容器之间的约定。约定包括组件的生命周期管理、实例用来从其容器中获取各种信息和服务的上下文接口、每个容器必须为其组件提供的服务列表。
component-managed sign-on （组件管理登录）	(n.)一种机制，其中登录资源所需的安全信息由应用程序组件提供。
component product descriptor file （组件产品描述符文件）	(n.)含有给定组件产品的元数据的文件（通常为 XML 格式）。
component state （组件状态）	(n.)描述日历事件（如会议）的属性集。在 WCAP 中， <code>compstate</code> 参数允许 <code>fetch</code> 命令按组件状态返回事件。例如， <code>compstate</code> 可能为 <code>REPLY-DECLINED</code> （参加者拒绝出席会议）或 <code>REQUEST_NEEDS-ACTION</code> （参加者在会议中尚未有任何行动）。
compromised key list （已泄密密钥列表）	参见 CKL 。
computed attribute （已计算属性）	(n.)一种属性，它不与条目本身一起存储而与操作结果中的常规属性一起返回到客户机应用程序。
conference room （会议室）	(n.)一种预先建立的聊天室，由管理员或其他拥有 <code>sysRoomsAdd</code> 权限的用户配置。管理员或其他拥有 <code>sysRoomsAdd</code> 权限的用户可以确定哪些用户可以查看和进入会议室。
configuration （配置）	(n.)调优服务器或为应用程序提供元数据的设置集合。通常，特定应用程序的配置保存在该应用程序的部署描述符文件中。另参见 admin console （管理控制台）、 deployment descriptor （部署描述符）。
configuration administrator （配置管理员）	(n.)拥有管理权限的人员，可管理服务器以及整个服务器软件拓扑中的配置目录数据。对于整个服务器软件拓扑中的所有资源，配置管理员拥有不受限的访问权。只有此管理员可向其他管理员分配服务器访问权限。在管理员组及其成员就位之前，由配置管理员负责管理性配置的初始管理。
Configuration Directory Server （配置目录服务器）	(n.)用于为一台或一组服务器维护配置信息的 Java Enterprise System Directory Server。
configuration file （配置文件）	(n.)包含服务器、应用程序或软件组件的配置参数的文件。
conflict （冲突）	(n.)在通过复制使两个服务器之间的数据保持同步之前，更改不同目录服务器上的相同目录数据时出现的情况。服务器同步时，将检测他们的副本是否一致，然后解决冲突或记录错误。

- conflict resolution (冲突解决方案)** (n.)用于解决更改信息的决定性程序。有关更多信息, 参见 Java Enterprise System Directory Server 管理指南。
- congestion thresholds (拥塞阈值)** (n.)由系统管理员设置的磁盘空间限制, 以便当系统资源不足时通过限制新操作来防止数据库过载。
- connection (连接)** (1) (n.)对于 **resource manager (资源管理器)**, 是代表与资源管理器之间的 **session (会话)** 对象。
(2) (n.)与 Java Enterprise System 消息服务器的活动连接。此连接可为队列连接或主题连接。
- connection factory (连接工厂)** (1) (n.)对于 **resource manager (资源管理器)**, 是用于创建资源管理器连接的对象。
(2) (n.)一种对象, 用于创建 Java 消息服务 (Java Message Service, JMS) 连接 (**TopicConnection** 或 **QueueConnection**), 应用程序代码通过这些连接可以使用所提供的 JMS 实现。应用程序代码使用 Java Naming and Directory Interface™ (JNDI) 服务查找采用 JNDI 名称的连接工厂对象。
- connection handler (连接处理器)** (n.)Directory Proxy Server 用来将收到的客户机请求分发给数据视图。系统根据收到的 IP 地址或域名等条件来将连接指定给连接处理器。处理连接时, 连接处理器会参考连接策略。
- connection policy (连接策略)** (n.)一种策略规则, 用于决定如何处理 Directory Proxy Server 连接处理器所路由的操作。资源限制策略限制了分配给连接、请求和引用的资源。请求过滤策略提供对数据的访问控制。
- connection pool (连接池)** (n.)一组连接。通过缓存和重新使用物理连接来避免连接开销, 以及减少大量线程之间共享的连接数量来实现数据库的高效访问。另参见 **JDBC connection pool (JDBC 连接池)**。
- connector (连接器)** (n.)容器的一种标准扩展机制, 用于提供到 EIS 的连接。连接器具有 EIS 特异性, 由资源适配器和获得 EIS 连接所需的应用程序开发工具组成。资源适配器通过支持连接器架构内定义的系统级订约而插入容器。另参见 **resource adapter (资源适配器)**。
- connector architecture (连接器架构)** (n.)用于集成 J2EE™ 应用程序与 EIS 的体系结构。此体系结构由两部分构成: EIS 供应商提供的资源适配器和允许此资源适配器插入的 J2EE 服务器。此体系结构定义了资源适配器为了能够插入 J2EE 服务器而必须支持的一组订约, 例如事务、安全性和资源管理。
- Connector for Microsoft Outlook** (n.)一种插件, 通过它可将 Microsoft Outlook 作为桌面客户端与 Sun Java Enterprise System 一起使用。
- console (控制台)** 参见 **admin console (管理控制台)**。
- consume (使用)** (v.)接收消息使用方从目的地带来的消息。
- consumer (使用方)** (1) (n.)包含从供应者服务器中复制的目录树或子树的服务器。
(2) (n.)由用于从目的地接收消息的会话创建的对象 (MessageConsumer)。在点对点传送模型下, 使用方就是一个接收器或浏览器 (**QueueReceiver** 或 **QueueBrowser**)。在发布/订阅传送模型下, 使用方就是一个订阅者 (TopicSubscriber)。

- consumer directory server (使用方目录服务器)** (1)(n.)一种只读目录服务器，引用对主目录服务器进行的所有添加、修改和删除操作。
- (2)(n.)任何从其他目录服务器接收更改的目录服务器。参见 [supplier directory server \(供应者目录服务器 \)](#)。
- contact (联系人)** (n.)向其发送即时消息以及从其接收即时消息的用户或 LDAP 组的 userID (名称)。可将联系人添加到个性化联系人组中，以便监视其在线状态。在其他即时讯息传递环境中也称作“好友”。
- contact group (联系人组)** (n.)用户维护的联系人列表。实际列表存储在 Instant Messaging Server 上。您可以创建联系人组以便合理地管理联系人。
- contact list (联系人列表)** (n.)在 Java Enterprise System Instant Messaging 中，它是所有联系人组的列表。
- container (容器)** (1)(n.)为特定类型的 J2EE 组件提供生命周期管理、安全性、部署和运行时服务。Application Server 为所有类型的 J2EE 组件提供容器。另参见 [component \(组件 \)](#)。
- (2)(n.)在 Java Enterprise System Portal Server 6.0 中，容器是指主要通过汇集其子频道内容的方法来生成自身内容的频道。在 Java Enterprise System Directory Server Access Management Edition 中，容器定义着获取其他 Directory Server Access Management Edition 对象的组织的类型。
- container entry (容器条目)** (n.)代表目录中子树顶层的条目。
- container-managed persistence (容器管理持久性)** (n.)一种机制，其中由实体 bean 的容器来管理 [entity bean \(实体 bean \)](#) 的变量和资源管理器之间的数据传输。另参见 [bean-managed persistence \(bean 管理持久性 \)](#)。
- container-managed relationship (容器管理关系)** (n.)一对类内字段间的关系，对此关系的一方进行的操作将会影响到另一方。
- container-managed sign-on (容器管理登录)** (n.)一种机制，其中登录资源所需的安全信息由容器提供。
- container-managed transaction (容器管理事务)** (n.)一种机制，其中企业 bean 的事务划分由 EJB 容器明确指定并自动控制。[entity bean \(实体 bean \)](#) 必须使用容器管理事务。另参见 [bean-managed transaction \(bean 管理事务 \)](#)。
- content (内容)** (n.)XML 文档中出现在 prolog 之后的部分，包括根元素及其包含的所有内容。
- context attribute (上下文属性)** (n.)限定到与 [servlet](#) 相关的上下文中的对象。
- context root (上下文根目录)** (n.)映射到 Web 应用程序的 [document root \(文档根目录 \)](#) 的名称。

- control descriptor (控制描述符)** (n.)一组企业 bean 配置条目, 使您能够指定可选单个属性覆盖 bean 方法, 以及企业 bean 事务和安全属性。
- controller (控制器)** (n.)与代理和存取程序组件交互的适用于 Windows 的身份认证服务连接器组件。控制器执行与密钥同步相关的任务, 例如确定用户在“同步用户列表”中的成员资格、搜索和链接对等的用户条目以及通过将当前用户条目与存储在对象高速缓存中的之前版本进行比较来检测对用户所做的更改。在操作的日志消息中经常会提及到控制器。
- conversational state (会话状态)** (1) (n.)对象状态由于与同一客户机的重复交互活动而更改。另参见 [persistent state \(持久状态\)](#)。
(2) (n.)[session bean \(会话 bean\)](#) 的字段值和可从 bean 字段获取的对象传递闭包。根据 Java 编程语言的序列化协议来定义 bean 的传递闭包 (即通过序列化 bean 实例来实现存储的字段)。
- cookie** (n.)信息的小集合, 可被发送到来访浏览器并在该浏览器后续的每一次访问时被检索, 以便服务器可识别来自同一客户机的访问。Cookies 具有域特异性, 并可利用与在应用程序和服务器之间进行其他数据交换相同的 Web 服务器安全功能。接受 cookies 可使 Web 页面加载更加快速, 同时不会对计算机带来安全威胁。
- cooperating server (协作服务器)** (n.)想与您的服务器进行通信以及您的服务器想与之通信的服务器。又称为“协同服务器”。每个协作服务器都有一个给定的符号名, 它是由字母和数字组成的字符串, 例如 `coservern`, 其中 n 为数字。
- CORBA** (common object request broker architecture, 公共对象请求代理体系结构) (n.)标准且独立于语言的体系结构定义, 用于 [OMG](#) 指定的面向对象的分布式计算。
- core service (核心服务)** (n.)用于定义 Java Enterprise System 服务器提供的基本功能的一个或多个关键服务, 与支持服务或附属服务相对。
- CoS** (class of service, 服务类) (n.)在条目间共享属性的方法。
- CoS definition entry (CoS 定义条目)** (n.)用于标识所用 CoS 的类型的条目。此条目作为 LDAP 子条目存储在其所影响的分支下。
- coserver (协同服务器)** 参见 [cooperating server \(协作服务器\)](#)。
- CoSNaming provider (CoSNaming 提供者)** (n.)为了支持全局 JNDI 名称空间 (IIOP 应用程序客户端可访问), Java Enterprise System Application Server 包括基于 J2EE 的 CosNaming 提供者, 它对 CORBA 引用 (远程 EJB 引用) 的绑定提供支持。
- CoSNaming Service (CoSNaming 服务)** (n.)基于 IIOP 的命名服务。
- CoS template entry (CoS 模板条目)** (n.)包含共享属性值的列表的条目。
- CRAM-MD5** (n.)RFC 2195 规范中记载的一种轻量标准跟踪验证机制。当仅需保护用户登录密码以防其遭到网络窃听时, 此机制可提供替代 TLS (SSL) 的快速 (虽然不太稳定) 备选措施。
- crawler** 参见 [robot](#)。

create method (create 方法)	(n.)在 home interface (home 接口) 中定义的方法，客户机调用它来创建 enterprise bean (企业 bean)。
CRL	(certificate revocation list , 证书撤销列表) (n.)由证书授权机构发布的列表，此列表用于指明客户机用户或服务器用户不应再信任的所有证书。在此情况下，该证书已被撤销。另参见 CKL 。
cronjob	(n.) (仅限 UNIX) 由时钟守护进程在配置过程中自动执行的任务。
CSAPI	(Calendar Server application programming interface , Calendar Server 应用编程接口) (n.)一种编程接口，提供修改或增强 Calendar Server 的功能集的能力。CSAPI 模块是在启动 Calendar Server 时从 <code>cal/bin/plugins</code> 目录加载的插件。
CSS	(1) (Cascading style sheet , 层叠样式表) (n.)一种与 HTML 和 XML 文件搭配使用的样式表，用于对使用特定标记标示的所有元素添加样式，从而定义浏览器或其他显示机制的呈现方式。
CTS	(Compatibility test suite , 兼容性测试套件) (n.)一种兼容性测试套件，用于检验 J2EE 产品是否符合 J2EE 平台规范。
CUA	(Calendar user agent , 日历用户代理) (n.)日历客户机用于访问 Calendar Server 的应用程序。

D

DAP	(directory access protocol, 目录访问协议) (n.)ISO/ITU-T X.500 协议, LDAP 的基础。
data (数据)	(n.)XML 流中的元素内容, 通常在元素不包含任何子元素时使用。当元素包含子元素时, 通常使用术语内容。当简单元素中仅包含 XML 结构的文本时, 以及当包含子元素的元素中混有少量数据或并未混有数据时, 通常将该结构看作 XML 数据 (与 XML 文档相对)。
data access logic (数据库访问逻辑)	(n.)涉及与某数据源的交互的业务逻辑。
database (数据库)	(n.)指代关系数据库管理系统 (relational database management system, RDBMS) 的通用术语。它是一种软件数据包, 可用于创建和处理大量相关的、经过组织的数据。另参见 schema (模式) 。
database connection (数据库连接)	(n.)与数据库或其他数据源之间的通信链接。各组件可同时创建和处理若干数据库连接, 以访问数据。
database wire protocol (数据库有线通信协议)	参见 data redundancy unit (数据冗余单元) 。
data redundancy unit (数据冗余单元)	(DRU) (n.)一组 HADB 节点, 其中含有半数的活动和备用节点以及一个完整的数据副本。HADB 被组织为两个 DRU, 这两个 DRU 互相镜像。为确保容错, 支持一个 DRU 的计算机在电源、处理单元和存储器方面必须完全自支持。另参见 HADB node (HADB 节点) 、 active node (活动节点) 、 spare node (备用节点) 及 mirror node (镜像节点) 。
data service (数据服务)	(n.)一种 Web 服务, 支持查询和修改与最终用户相关的数据。数据服务示例是托管和公开用户的配置文件信息 (例如名称、地址和电话号码) 的 Web 服务。
data source (数据源)	(1) (n.)数据源 (例如数据库) 的句柄。数据源在 Application Server 注册, 然后以编程方式检索, 以与数据源建立连接和接口。数据源定义指定连接数据源的方式。 (2) (n.)通过 Directory Proxy Server 访问的系统信息库。系统信息库包括 LDAP 目录、与 JDBC 兼容的数据库以及 LDIF 平面文件。
data source object (数据源对象)	(n.)数据源对象具备一组属性, 用于标识和描述此对象代表的实际数据源。

data source pool (数据源池)	(n.)一组拥有等效数据的数据源。数据源池为 Directory Proxy Server 提供负载平衡和故障转移管理。
data store (数据存储)	(1) (n.)包含目录信息的存储区, 通常用于整个 DIT。 (2) (n.)永久存储 Message Queue 代理所需信息 (持久订阅、关于目的地的数据、持久性消息、审计数据) 的数据库。
data view (数据视图)	(n.)使用基于 DN 的路由选择将连接从 Directory Proxy Server 连接处理器路由至数据源池。
DC tree (DC 树)	(domain component tree, 域组件树) (n.)镜像 DNS 网络语法的一种 DIT。“DC 树”中的标识名示例为 cn=billbob,dc=bridge,dc=net,o=internet。
DDP	(Document-driven programming, 文档驱动编程) (n.)使用 XML 定义应用程序。
declaration (声明)	(n.)XML 文档的头等大事, 即将其声明为 XML。最小声明为 <?xml version="1.0"?>。声明为文档 prolog 的一部分。
declarative security (声明性安全)	(n.)组件部署描述符中的声明性安全属性, 允许组件的容器 (例如, bean 的容器或 servlet 引擎) 隐式管理安全性。这种安全性不需要程序控制。与 programmatic security (程序安全性) 相对。另参见 container-managed persistence (容器管理持久性)。
declarative transaction (声明性事务)	参见 container-managed transaction (容器管理事务)。
decryption (解密)	(n.)破解加密信息的过程。另参见 encryption (加密)。
default calendar (默认日历)	(n.)用户登录到 Calendar Express 之后看到的第一个日历。默认日历的日历 ID 通常与用户的用户 ID 相同。例如, jdoe@example.com 的默认日历名称为 jdoe。
default index (默认索引)	(n.)安装 Directory Server 时为每个数据库实例创建的一组索引。安装 Java Enterprise System Directory Server 之后, 即为每个数据库实例创建了一组默认索引。有关更多信息, 参见 Java Enterprise System Directory Server 管理指南。
defederation (取消联合)	(n.)参见 federation termination (联合终止)。
definition entry (定义条目)	参见 CoS definition entry (CoS 定义条目)。
defragmentation (取消分段)	(n.)MIME 功能, 可使分为若干短消息或片段的长消息重新组合。每一段中显示的 "Message/Partial" Content-Type 标头字段所含的信息有助于将各段重新组合为一条消息。另参见 fragmentation (分段)。
Delegated Administrator	(n.)一组 GUI 和 CLI 界面, 它们使管理员能够向托管域添加用户并修改托管域中目录的用户和组。
delegated administrator console (委托管理员控制台)	(n.)一种基于 Web 浏览器的软件控制台, 使域管理员可以为托管域添加和修改用户及组。还允许最终用户更改其密码、设置消息转发规则、设置休假规则和列出邮件列表订阅。

delegated administrator for messaging and collaboration (消息传送和协作委托管理员)	(n.)使域管理员可以对托管域添加和修改用户及用户组的一组界面 (GUI 和实用程序) 。
delegated administrator server (委托管理员服务器)	(n.)采用托管域处理对目录的访问控制的守护程序。
delegation (委托)	(1) (n.)一种面向对象的技术, 将若干对象的组合作为一种实现策略使用。负责操作结果的一个对象将实现过程委托给另一对象。例如, 类加载器经常将某些类的加载委托给其父项。另参见 class loader (类加载器) 。
delete a message (删除消息)	(v.)标记某消息以进行删除。删除的消息不会从消息存储中实际移除, 除非用户通过单独操作将其擦除或清除。另参见 purge a message (清除消息) 、 expunge a message (清除邮件) 。
delivery (传送)	参见 message delivery (消息传送) 。
delivery mode (传送模式)	(n.)表示消息传送可靠性的一种模式: 消息可被保证传送并成功使用一次且仅使用一次 (持久性传送模式) , 或者被保证至少传送一次 (非持久性传送模式) 。
delivery model (传送模型)	(n.)传送消息时利用的模型。模型可为点对点或发布/订阅式。在 Java™ 消息服务 (Java Message Service, JMS) 中, 每一模型都有单独的编程域, 这些域使用特定的客户机运行时对象和目的地类型 (队列或主题) 以及一个统一的编程域。
delivery policy (传送策略)	(n.)一种规范, 详细说明注册多个消息使用方后队列如何路由消息。此类策略有单一、故障转移和循环。
delivery status notification (传送状态通知)	(n.)此类消息提供向某收件人发送的消息的状态信息, 例如指明由于网络故障而延迟传送的消息。
denial of service attack (拒绝服务攻击)	(n.)个人通过向邮件服务器发送大量消息有意或无意对其进行攻击的一种情形。服务器的吞吐量可能会受到严重影响, 服务器本身也可能过载和失效。
deny filter (拒绝过滤器)	(n.)Java Enterprise System Messaging Server 的一种访问控制规则, 用于确定被拒绝访问以下一项或多项服务的客户机: POP、IMAP 或 HTTP。另参见 Allow filter (允许过滤器) 。
deployer (部署者)	(n.)将 J2EE 模块和应用程序安装到操作环境中的人员。
deployment (部署)	(1) (n.)将软件安装到操作环境中的过程。 (2) (n.)Java Enterprise System 解决方案生命周期过程中的一个阶段, 在此阶段, 部署方案被转化成部署设计, 后者被实现、原型化而后在生产环境中铺开。此过程的最终产品也被称为部署 (或部署的解决方案) 。
deployment architecture (部署体系结构)	(n.)一种高层次设计, 描述了 logical architecture (逻辑体系结构) 到物理计算环境的映射。物理环境包括内联网或 Internet 环境中的计算机、它们之间的网络链路以及支持软件所需的其他物理设备。

- deployment descriptor (部署描述符)** (n.)随各模块和应用程序提供的一种 XML 文件, 用于描述应如何部署各应用程序。部署描述符指示部署工具通过特定的容器选项部署某模块或应用程序, 并说明了部署者必须解决的特定配置要求。另参见 [metadata \(元数据\)](#)。
- deployment scenario (部署方案)** (n.)用于 Java Enterprise System 解决方案的一种 [logical architecture \(逻辑体系结构\)](#) 以及解决方案要满足业务需求而必须达到的服务质量要求。服务质量要求包括下列各方面的要求: 性能、可用性、安全性、可维护性以及可伸缩性/潜在容量。部署方案是部署设计的起点。
- depth (深度)** (n.)从“搜索引擎”中某站点的起点开始追随的链接的数量。在您定义站点时, 即定义 robot 从该点开始可追随的链接的数量, 从而限定搜索的深度。
- dereference an alias (解除别名引用)** (v.)在绑定或搜索操作中, 指定目录服务将某个条目的别名标识名转换为实际标识名。
- Desktop (桌面)** 参见 [Portal Server Desktop](#)。
- destination (目的地)** (n.)Java Enterprise System 消息服务器中的物理位置, 生成的消息被传送到此位置进行路由选择, 随后传送给使用方。该物理目的地 ([queue \(队列\)](#) 或 [topic \(主题\)](#)) 由 [administered object \(受管对象\)](#) 标识和封装。客户机使用此受管对象指定目的地, 客户机为此目的地生成消息并从此目的地使用消息。另参见 [point-to-point delivery model \(点对点传送模型\)](#)、[publish and subscribe delivery model \(发布和订阅传送模型\)](#)。
- destination resource (目的地的资源)** (n.)代表“主题”目标或“队列”目的地的对象。由各应用程序用来对“队列”进行读写操作或对“主题”进行发布和订阅。应用程序代码使用 Java Naming and Directory Interface™ (JNDI) 服务查找使用 JNDI 名称的 Java 消息服务 (Java Message Service, JMS) 资源对象。
- development (开发)** (n.)Java Enterprise System 解决方案部署过程中的一项任务, 该任务可对 [deployment architecture \(部署体系结构\)](#) 的自定义组件进行编程和测试。
- device detection (设备检测)** 参见 [client detection \(客户机检测\)](#)。
- device information (设备信息)** (n.)用于 Portal Server Mobile Access 的设备特定客户机数据。
- DHCP** (dynamic host configuration protocol, 动态主机配置协议) (n.)Internet 提出的一种标准协议, 该协议允许系统将 IP 地址动态分配给网络上的各个计算机。另参见 [IP address \(IP 地址\)](#)。
- digest authentication (摘要验证)** (n.)一种验证类型, 允许用户无需以明文方式发送用户名和密码即可进行验证。Web 应用程序通过向服务器发送消息摘要和其 HTTP 请求消息来向 Web 服务器验证自身。通过对一连串的 HTTP 请求消息和客户机密码应用单向散列算法 (称为 MD5) 来计算摘要。摘要通常比 HTTP 请求小得多, 且不包含密码。服务器使用“摘要验证”插件比较客户机提供的摘要值。
- DIGEST-MD5** (n.)一种比 CRAM-MD5 更安全的轻量标准跟踪验证机制。记载于 RFC 2831 规范中, 同时提供了无需额外的 TLS (SSL) 设置开销即可保护整个连接的选择。

digital signature (数字签名)	(n.)用于验证消息和签名者的一种电子安全机制。
directive (指令)	(n.)一种“搜索引擎”语句，它使用特殊格式调用操作程序（例如 robot 应用操作程序）并将参数传递给参数块中的操作程序。例如，以下指令调用 <code>enumerate-urls</code> 函数，并为 <code>max</code> 和 <code>type</code> 传递参数： <code>Enumerate fn=enumerate-urls max=1024 type=text/html</code>
directory (目录)	(n.)一种经过优化的特殊数据库，优化目的是为了读取数据而不是为了写入数据。大多数目录都基于行业标准协议 LDAP（轻型目录访问协议）。
directory access protocol (目录访问协议)	参见 DAP 。
directory context (目录上下文)	(n.)目录信息树中的点，搜索用于验证消息存储访问的用户名和密码的条目时，该点为一个起点。另参见 base DN (基 DN) 。
directory deployment (目录部署)	(n.)在 Application Server 中，以展开目录而非归档文件的形式部署未打包的 J2EE 应用程序或模块。
directory entry (目录条目)	(n.)由标识名标识的一组目录属性及它们的值。每个条目都包含一个对象类属性，说明条目描述的对象种类并定义条目包含的一组属性。
directory information tree (目录信息树)	参见 DIT 。
directory lookup (目录查找)	(n.)根据指定用户或资源的名称或其他特征搜索目录，以查找有关该用户或资源的信息的过程。
Directory Manager (目录管理员)	(1) (n.)对目录服务器数据库具备管理权限的用户。该用户不具备访问控制权限（将目录管理员视为目录的超级用户）。 (2) (n.)具备特权的数据库管理员，类似于 UNIX 系统中的超级用户。目录管理员不具备访问控制权限。
directory schema (目录模式)	(n.)定义可存储在目录中的数据的一组规则。
Directory Server	(n.) Lightweight Directory Access Protocol (轻量目录访问协议) 的 Java Enterprise System 版本。Application Server 的每一个实例都使用 Directory Server 来存储共享的服务器信息，包括用户和组的有关信息。
Directory Server Access Management Edition	(n.)一组界面，为 Portal Server 提供用户和服务管理、验证和单点登录服务、策略管理、日志记录服务、调试实用程序和客户机支持。
directory server (目录服务器)	(1) (n.)通过逻辑集中式信息库提供某组织内人员和资源的相关信息的服务器。另参见 LDAP 和 Directory Server Access Management Edition (2) (n.)基于 LDAP 的 Java Enterprise System 目录服务。
directory service (目录服务)	(n.)一种数据库应用程序，用于管理组织内人员和资源的基于属性的描述性信息。

Directory Service Control Center	(n.)基于浏览器的 GUI，用于管理 Directory Server 和 Directory Proxy Server。
Directory Service Manager (目录服务管理员)	(n.)通过 Directory Service Control Center 管理多个 Directory Server 和 Directory Proxy Server 上的服务器配置和数据的 LDAP 超级用户。
directory synchronization (目录同步)	(n.)将 MTA 目录高速缓存与存储在目录服务中的当前目录信息同步的过程。另参见 MTA directory cache (MTA 目录高速缓存) 。
disconnected state (断开状态)	(n.)在此状态下，邮件客户机连接至服务器，创建选定消息的高速缓存副本，然后断开与服务器的连接。
Dispatcher (分发程序)	(n.)为已定义的 TCP 端口处理连接请求的 MTA 组件。“分发程序”是一种多线程连接分发代理程序，允许多个多线程服务器共同负责分担某项指定服务。使用“分发程序”时，您可以使若干多线程 SMTP 服务器进程同时运行。
display profile (显示配置文件)	(n.)一组 XML 文档，用于在 Java Enterprise System Portal Server 中定义和配置提供者和频道。
distinguished name (标识名)	参见 DN 。
distributable session (可分布会话)	(n.)可在某个群集的所有服务器中分布的用户会话。
Distributed Authentication UI Server	(n.)一种 Access Manager 子组件，它可跨过 Access Manager 部署中的两道防火墙提供安全的分布式验证。请将 Distributed Authentication UI 子组件安装在位于 Access Manager 部署非安全 (DMZ) 层内的一个或多个服务器上。此子组件充当最终用户与第二道防火墙后的 Access Manager 实例之间的验证接口，从而避免了将 Access Manager 服务 URL 直接暴露给最终用户。
distributed enterprise application (分布式企业应用程序)	(n.)一种应用程序，其逻辑遍布于整个网络或 Internet 环境（分布式方面），并且其范围和规模满足了生产环境或服务提供者的需要（企业方面）。应用程序的组件运行于单独的运行时环境（通常在不同的平台上）中。典型的分布式应用程序有双层（客户机-服务器）、三层（客户机-中间件-服务器）和多层（客户机-多个中间件-多个服务器）结构。
distributed indexing (分布式索引建立)	(n.)在“搜索引擎”中分配不同 robot 以为网络的不同部分建立索引的过程。采用分布式索引建立可减少每个 robot 的负载。这样，一个单独的“搜索引擎”便可从所有不同的 robot 收集所有资源说明，方法是导入每个 robot 的资源说明。
distributed transaction (分布式事务)	(n.)可应用于多个可能驻留在各个单独服务器上的异构数据库的单独事务。
distribution list (分发列表)	参见 mail list (邮件列表) 。
distribution list owner (分发列表所有者)	参见 mail list owner (邮件列表所有者) 。

DIT	(directory information tree, 目录信息树) (n.)存储在目录中的信息的逻辑表示法。DIT 镜像多数文件系统使用的树模型, 树根点显示在分层结构的顶部。
DN	(distinguished name, 标识名) (n.)目录中条目名称和位置的字符串表示法。
DN attribute (DN 属性)	(n.)包含关联用户、用户组或对象的标识信息的文本字符串。
DNS	(domain name system, 域名系统) (n.)网络中的计算机用来将 IP 地址 (例如 00.120.000.168) 与主机名 (例如 www.example.com) 相关联的系统。客户机通常使用 DNS 查找它们要联系的服务器的 IP 地址。DNS 中的数据通常在本地表表中扩充, 例如从 NIS 或 UNIX 系统中的 <code>/etc/hosts</code> 文件。另参见 IP address (IP 地址) 。
DNS alias (DNS 别名)	(n.)DNS 服务器了解的指向另一主机的主机名。DNS 别名作为一个 DNS CNAME 记录执行。计算机始终具备一个实名, 但它们可有一个或多个别名。例如, 别名 www.example.com 可能指向目前充当服务器的名为 realthing.example.com 的实际计算机。
DNS database (DNS 数据库)	(n.)域名 (主机名) 及其相应的 IP 地址的数据库。
DNS domain (DNS 域)	(n.)一组计算机, 它们的主机名共用一个公用后缀 (域名)。从语句构成上说, 一个 Internet 域名由一系列句点 (圆点) 分隔的名称 (标签) 构成, 例如 corp.mktng.example.com 。另参见 domain (域) 。
DNS spoofing (DNS 电子欺骗)	(n.)一种网络攻击形式, 在这种攻击下, DNS 服务器遭到破坏, 因而提供虚假的信息。
document (文档)	(1) (n.)网络中的一种文件, 多数是网页或字处理文档, 但也可能是文本文件、电子表格等等。按“搜索引擎”建立索引的资源的通用术语。 (2) (n.)一种 XML 结构, 其中有一个或多个元素包含混合了子元素的文本。另参见 data (数据) 。
Document Object Model, DOM (文档对象模型)	(n.)一种 API, 用于访问 XML 文档并将其处理为树结构。DOM 提供与平台和语言无关的接口, 使程序和脚本可动态访问与修改 XML 文档中的内容及结构。
document root (文档根目录)	(1) (n.)服务器上的目录, 此目录包含的文件、图像和数据将显示给访问 Java Enterprise System Web Server 的用户。 (2) (n.)服务器上的目录, 此目录包含您要显示给访问该服务器的用户的文件、图像和数据。 (3) (n.)文档根目录 (有时称为文档主目录) 是包含您要提供给远程客户机的所有虚拟服务器文件的中心目录。
document type definition (文档类型定义)	参见 DTD 。
domain (域)	(1) (n.)全限定域名的最后一部分, 用于标识拥有此域名的公司或组织 (例如, example.com 、 host.example.com)。 (2) (n.)受某单独计算机系统管理控制的资源。

- (3) (n.)Java 消息服务 (Java Message Service, JMS) 客户端用来为 JMS 消息传送操作编程的一组对象。有两个编程域：一个用于点对点式传送模型，一个用于发布/订阅式传送模型。
- (4) (n.)Sun Java System Application Server 内的一个功能，允许不同的管理用户创建和管理自己的域。域是在单个系统中使用已安装的一组公用二进制文件创建的一组实例。
- Domain Administration Server (域管理服务器)** (n.)“域管理服务器”是一个专门指定的 Application Server 实例，用于为 Application Server 处理所有管理任务。它可维护和更新 Application Server 配置信息的中心信息库。如果“域管理服务器”未在运行，则无法执行管理任务。
- domain administrator (域管理员)** (n.)拥有以下管理权限的用户：可以使用消息传送和协作 GUI 或 CLI 的“委托管理员”创建、修改、删除托管域中的邮件用户、邮件列表及家庭帐户。默认情况下，此用户可以担任拓扑中所有讯息传递服务器的消息存储管理员。
- domain alias (域别名)** (n.)指向另一域的域条目。使用别名使托管域可拥有若干个域名。
- domain directory (域目录)** (n.)Application Server **domain (域)** 的目录，至少包含一个 **instance directory (实例目录)**。它是 **server root (服务器根目录)** 在 Application Server 中的叫法。
- domain hosting (域托管)** (n.)托管域的过程。在共享讯息传递服务器上托管一个或多个域的功能。例如，域 `example.com` 和 `example.org` 均可托管在 `example.com` 邮件服务器上。用户向此托管域发送邮件，并从那里接收邮件。邮件服务器的名称不出现在邮件地址中。
- domain name (域名)** (1) (n.)在电子邮件地址中使用的主机名。
(2) (n.)定义管理组织的唯一名称。域可包含其他域。域名的解释顺序为从右至左。例如，`example.com` 既是 Example 公司的域名，又是顶级 `com` 域的子域。`example.com` 域可进一步分为子域，例如 `corp.example.com` 等等。另参见 **host name (主机名)**、**fully qualified domain name (全限定域名)**。
- domain name system (域名系统)** 参见 **DNS**。
- domain organization (域组织)** (n.)组织树中托管域下方的子域。对于要按部门编排用户和用户组条目的公司，域组织很有用。
- domain part (域部分)** (n.)电子邮件地址中“at”符号 (@) 右侧的部分。例如，`example.com` 是电子邮件地址 `jdoe@example.com` 的域部分。
- domain quota (域配额)** (n.)分配给电子邮件消息的域的空间数量。空间数量由系统管理员配置。
- domain registry (域注册表)** (n.)一种单独的数据结构，其中包含安装服务器时创建和配置的所有域的域特定信息（例如域名、域位置、域端口、域主机）。
- domain rewrite rules (域重写规则)** 参见 **rewrite rule (重写规则)**。
- domain template (域模板)** (n.)重写规则的一部分，定义如何重写地址的主机和域部分。此模板可包括一个完全静态的主机和域地址或一个单独的字段替代字符串，或两者都包括。

double failure (双故障)	(n.)HADB 中一个或多个镜像节点对同时发生故障。参见 HADB 、 HADB node (HADB 节点) 、 active node (活动节点) 、 spare node (备用节点) 、 mirror node (镜像节点) 和 data redundancy unit (数据冗余单元) 。
drop word (丢弃词)	参见 stop word (停用词) 。
DRU	参见 data redundancy unit (数据冗余单元) 。
DSA	(directory system agent, 目录系统代理) (n.)Directory Server 的 X.500 术语。
DSCC	参见 Directory Service Control Center 。
DSE	(directory server entry, 目录服务器条目) (n.)一种条目或特定于 DSA 的条目, 含有与其相关的特定于服务器的附加信息。DSE (如根 DSE 或模式 DSE) 在每个服务器上具有不同的属性。
DSP	(digital signal processing, 数字信号处理) (n.)模拟信号到数字信号的转换。使用电话以语音方式访问 Portal Server 软件需要 DSP 卡。
DSML	(directory services markup language, 目录服务标记语言) (n.)一组表示 XML 标记语言的文档格式, 通过它们可用 XML 表示目录服务。Java Enterprise System Directory Server 5.2 符合 DSML 标准的版本 2 (DSMLv2)。
DSN	(n.)参见 delivery status notification (传送状态通知) 。
dsservd	(n.)一种守护进程, 它访问存放目录信息的数据库文件, 并使用 LDAP 协议与目录客户机通信。
dssetup	(n.)一种 Java Enterprise System Directory Server 准备工具, 它可以使现有的 Directory Server 做好供 Java Enterprise System Messaging Server 使用的准备。
DTD	(document type definition, 文档类型定义) (n.)XML 文档 prolog 的可选部分 (如 XML 标准所指定)。DTD 约束可在文档中使用的有效标记和标记序列。DTD 存在一些缺点, 不过它也产生了各种模式提案。例如, DTD 条目 <code><!ELEMENT username (#PCDATA)></code> 表示称为 username 的 XML 元素包含已解析的字符数据, 即下面不包括任何其他结构元素的纯文本。DTD 包含在当前文件中定义的本地子集和外部子集, 外部子集由在本地子集中使用参数实体来引用的外部 DTD 文件中所包含的定义构成。
durable subscription (持久订阅)	(n.)在 JMS publish and subscribe delivery model (发布和订阅传送模型) 中, 无论当前是否存在活动的订阅者对象都会继续存在的订阅。如果没有活动的订阅者, JMS provider (JMS 提供者) 会保留订阅的 message (消息) , 直到它们被订阅接收或过期。
DWP	(database wire protocol, 数据库有线通信协议) (n.)一种 Calendar Server 专有协议, 它允许多个服务器在同一 Calendar Server 系统内链接在一起, 以形成分布式日历存储。Calendar Server 使用 DWP 检索存储在日历数据库中的远程数据。
dynamic deployment (动态部署)	(n.)在 Application Server 中, 系统会动态部署或重新部署 J2EE 应用程序或模块, 也就是无需重新启动服务器。另参见 dynamic reloading (动态重新加载) 。
dynamic group (动态组)	(n.)由 LDAP 搜索 URL 定义的邮件组。用户通常通过在其目录条目中设置 LDAP 属性加入组。

dynamic reloading (动态重新加载) (n.)修改和重新加载之前部署的组件的过程，该过程无需完成整个部署过程，也无需重新启动服务器。默认情况下，可以动态地重新加载 servlet、使用 JavaServer Pages™ 技术 ([JSP technology \(JSP 技术\)](#)) 创建的页面以及企业 bean 组件。另参见 [dynamic deployment \(动态部署 \)](#)。

dynamic web application (动态 Web 应用程序) (n.)指 servlet、JSP™ 页面、内容提供者或任何其他需要由 Java Web 容器处理的程序，该容器通过用户浏览器访问。对于 Java Enterprise System Portal Server，该应用程序安装在 Web 服务器中。

E

- EAR file (EAR 文件)** (enterprise archive file, 企业归档文件) (n.)含有 J2EE 应用程序的归档文件。EAR 文件带有 .ear 扩展名。
- ebXML** (Electronic Business XML, 电子商务 XML) (adj.)一组规范, 旨在使企业通过交换基于 XML 的消息来开展业务。它是由 OASIS (结构化信息标准促进组织) 和联合国贸易便利与电子商务中心 (United Nations Centre for the Facilitation of Procedures and Practices in Administration, Commerce and Transport, U.N./CEFACT) 发起的。
- ebXML registry (ebXML 注册表)** (Electronic Business XML registry, 电子商务 XML 注册表) (n.)一种联合 registry (注册表) 和系统信息库, 管理各类由标准和可扩展 metadata (元数据) 描述的电子内容。
- ECC** (elliptic curve cryptography, 椭圆曲线密码学) (n.)一种用于移动或无线环境且基于椭圆曲线数学的公钥密码学。
- e-commerce** (electronic commerce, 电子商务) (n.)表示通过 Internet 开展业务的术语。
- EHLO command (EHLO 命令)** (n.)SMTP 命令。该命令对服务器进行查询以确定该服务器是否支持 SMTP 扩展命令。在 RFC 1869 中定义。
- EIS** (enterprise information system, 企业信息系统) (n.)构成企业现有系统的应用程序, 用于处理全公司的信息。这些应用程序为企业提供信息基础结构。企业信息系统为其客户机提供一组明确定义的服务。以 local 接口或 remote 接口或二者皆有的形式向客户机展示这些服务。企业信息系统示例包括企业资源计划系统、大型事务处理系统和传统数据库系统。特定示例有 R/3、PeopleSoft、Tuxedo 和 CICS。
- EIS resource (EIS 资源)** (n.)一种向其客户机提供特定于企业信息系统的功能的资源。示例包括数据库系统中的一个记录或一组记录、企业资源计划系统中的业务对象、事务处理系统中的事务程序。
- EJB container (EJB 容器)** (n.)一种实现 J2EE 体系结构的 EJB 组件约定的容器。此约定指定 enterprise bean (企业 bean) 的运行环境, 包括安全性、并发性、生命周期管理、事务、部署、命名和其他服务。EJB 容器由 EJB 或 J2EE 服务器提供。另参见 container (容器)。
- EJB container provider (EJB 容器提供者)** (n.)提供 EJB 容器的供应商。

- EJB context (EJB 上下文)** (n.)一种对象, 允许 **enterprise bean (企业 bean)** 调用容器提供的服务以及获取有关客户机调用方法的调用方的信息。
- EJB home object (EJB 主对象)** (n.)一个为 **enterprise bean (企业 bean)** 提供生命周期操作 (创建、删除、查找) 的对象。EJB 主对象的类由容器的部署工具生成。EJB 主对象实现企业 bean 的 **home interface (home 接口)**。客户机引用 EJB 主对象来对 EJB 对象执行生命周期操作。客户机使用 **JNDI name (JNDI 名称)** 来查找 EJB 主对象。
- EJB JAR file (EJB JAR 文件)** (n.)包含 **EJB module (EJB 模块)** 的归档文件。EJB JAR 文件具有 .jar 扩展名。
- EJB module (EJB 模块)** (n.)一种可部署单元, 由一个或多个 **enterprise bean (企业 bean)** 和 EJB 部署描述符组成。另参见 **module (模块)**。
- EJB object (EJB 对象)** (n.)一种对象, 它的类实现 **enterprise bean (企业 bean)** 的 **remote interface (remote 接口)**。客户机不会直接引用企业 bean 实例; 而是始终引用 EJB 对象。EJB 对象的类由容器的部署工具生成。
- EJB server (EJB 服务器)** (n.)向 **EJB container (EJB 容器)** 提供服务的软件。例如, EJB 容器通常依靠作为 EJB 服务器的一部分的事务管理器来在所有参与的资源管理器中分两个阶段执行提交。J2EE 体系结构假设 EJB 容器是由来自同一供应商的 EJB 服务器托管的, 因此它不会指定这两个实体之间的订约。EJB 服务器可托管一个或多个 EJB 容器。
- EJB server provider (EJB 服务器提供者)** (n.)提供 EJB 服务器的供应商。
- EJB™ QL** (EJB Query Language, EJB 查询语言) (n.)为具有 **container-managed persistence (容器管理持久性)** 的 **entity bean (实体 bean)** 的 **finder** 和 **select** 方法定义查询。作为 SQL92 的子集, EJB QL 还进行了一些扩充以在实体 bean 的 **abstract schema (抽象模式)** 中定义的关系上进行导航。
- EJB technology (EJB 技术)** (Enterprise JavaBeans™ technology, 企业 JavaBeans 技术) (n.)一种组件体系结构, 用于开发和部署面向对象的分布式企业级应用程序。使用企业 JavaBeans 体系结构编写的应用程序具有可伸缩性、事务性和安全性。另参见 **enterprise bean (企业 bean)**。
- ejbc utility (ejbc 实用程序)** (n.)企业 bean 的编译程序。此实用程序检查所有 EJB 类和接口是否符合 EJB 规范, 并生成桩模块和框架。
- element (元素)** (n.)一个较大集合的成员, 例如一个数组中的数据单元或一个逻辑元素。在 XML 文件中, 元素是基本的结构单元, 并由标记进行定义。XML 元素含有子元素或 **data (数据)**, 并且可能包含 **attribute (属性)**。
- elliptic curve cryptography (椭圆曲线密码学)** 参见 **ECC**。
- empty tag (空标记)** (n.)不包括任何内容的 **XML** 标记。
- encryption (加密)** (n.)通过使信息无法识读而保护信息免受未授权使用的过程。有些加密方法使用对信息进行加密的代码, 称为密钥。另参见 **decryption (解密)**。
- endpoint (端点)** (1) (n.)负载均衡群集中某机器的 IP 地址或主机名。

	(2)(n.)在 Java 消息服务中, 为消息使用方。参见 message-driven bean (消息驱动 bean) 。
	(3)(n.)一种 Java 类, 通常为 <code>Servlet</code> 或无状态会话 <code>bean</code> , 使用 <code>javax.jws.WebService</code> 注释进行注解。该注释将类定义为从 Web 服务客户端接收消息的 web service (Web 服务) 端点。
end user (最终用户)	(n.)常常通过图形用户界面 (如 Internet 浏览器或移动设备 GUI) 使用分布式应用程序的人员。应用程序同时支持的最终用户数是决定应用程序的 deployment architecture (部署体系结构) 的一个重要因素。
ENS	参见 event notification service (事件通知服务) 。
enterprise bean (企业 bean)	(n.)实现业务任务或业务实体的 J2EE 组件, 由 EJB 容器托管; 可以是 entity bean (实体 bean) 、 message-driven bean (消息驱动 bean) 或 session bean (会话 bean) 。另参见 container (容器) 。
enterprise bean provider (企业 bean 提供者)	(n.)应用程序开发者, 负责生成企业 <code>bean</code> 类、 <code>remote</code> 和 <code>home</code> 接口、部署描述符文件, 并将它们打包到 EJB JAR 文件中。
enterprise network (企业网络)	(n.)由相互连接的网络集合构成的网络, 这些网络集合分布在一个地理上分散的区域中。企业网络满足了分布广泛的公司的需要, 并用于公司的关键任务应用程序。
entity (实体)	(1)(n.)在 XML 文件中, 实体为一个独特的单个项, 可通过引用的方式将其包括在 XML 文档中。此类实体引用可将实体指定为与字符一般大小 (例如 <code>&lt;</code> ; 它引用小于符号或左尖括号 <code><</code>)。实体引用也可以引用整个文档、外部实体或 DTD 定义的集合。
entity bean (实体 bean)	(n.)EJB 1.x 或 2.x enterprise bean (企业 bean) , 它代表在数据库中维护的持久性数据。实体 <code>bean</code> 可管理自身的持久性, 也可将此功能委托给其容器。实体 <code>bean</code> 由主键标识。如果托管实体 <code>bean</code> 的容器发生崩溃, 此实体 <code>bean</code> 、其主键以及所有远程引用都能免于崩溃。实体 <code>bean</code> 总是事务性的并可被多用户识别。另参见 persistence (持久性) 、 message-driven bean (消息驱动 bean) 、 read-only bean (只读 bean) 和 session bean (会话 bean) 。
entity reference (实体引用)	(n.)解析 XML 文档时替代引用的实体引用。它可以引用预定义实体 (如 <code>&lt;</code>) 或引用在 DTD 中定义的实体。在 XML 数据中, 引用可以引用 DTD 的本地子集中定义的实体或外部 XML 文件 (外部实体)。DTD 也可划出 DTD 规范的某个段并为其命名, 这样就可以通过定义参数实体以在 DTD 中的多个点上重复使用 (包括) 该段。
entropy (熵)	(n.)封闭系统中随机性的量度。特别是在 SSL 上下文中, 为了在随机数生成中引入熵 (确保随机性), 使用了多个种子。
entry (条目)	(n.)一组属性和一个独有的标识名。
entry distribution (条目分发)	(n.)跨多个服务器分发目录条目的方法, 以便扩展为可支持大量条目。
entry ID list (条目 ID 列表)	(n.)条目 ID 的列表。目录使用的每个索引由一个索引键表和相应的条目 ID 列表组成。目录使用条目 ID 列表构建可能符合客户端应用程序搜索请求的备选条目的列表。
enumeration (枚举)	(n.)robot 操作的一个阶段。在此阶段中, robot 寻找资源, 包括提取和执行超文本链接。
envelope (信封)	(n.)含有关于电子邮件消息的发件人和收件人传输信息的容器。此信息并非消息头的组成部分。在消息从一个地点传输到另一个地点时, 各种电子邮件程序都要使用信封。用户只能看到消息头和消息正文。

envelope field (信封字段)	(n.)消息信封中已命名的信息项, 如 RCPT TO。
equality index (等同索引)	(n.)一种索引, 利用它可以有效地搜索包含特定属性值的条目。
ERP	(enterprise resource planning, 企业资源计划) (n.)一种多模块软件系统。该系统通常包括一个关系数据库和多个应用程序, 用以管理采购、库存、人员、客户服务、运输、财务计划和业务的其他重要方面。
error handler (错误处理程序)	(n.)处理错误的程序。在 Messaging Server 中, 错误处理程序发出出错消息, 并在邮寄主管填好错误处理程序操作表单后, 对其进行处理。
error handler action form (错误处理程序操作表单)	(n.)发送到邮寄主管帐户的表单。该表单伴有 Messaging Server 不能处理的收到消息。邮寄主管通过填写表单指示服务器如何处理消息。
ESMTP	参见 extended simple mail transfer protocol (扩展的简单邮件传输协议) 。
ESP	(n.)enterprise service provider (企业服务提供者)。
ETRN command (ETRN 命令)	(n.)SMTP 命令, 该命令可以使客户机请求服务器启动其邮件队列的处理, 以便获得在服务器等待客户机的消息。在 RFC 1985 中定义。
event (事件)	(1) (n.)在日历中具有关联日期和时间的条目。例如, 事件可以是日历中的一个新会议或日程。 (2) (n.)一个已命名的操作。该操作触发模块或外部 Java Naming and Directory Interface™ (JNDI) 资源的响应。 (3) (n.)受管理对象在状态、控制权、严重性或描述方面发生的变化。 (4) (n.)在 Application Server 中, 触发与服务器自我管理规则相关的操作的状况。另参见 management rule (管理规则) 。
event notification service (事件通知服务)	(n.)一种通用服务。该服务接受可分类的服务器级别事件的报告, 然后通知其他已登记对某些类别事件感兴趣的服务器。允许 Java Naming and Directory Interface (JNDI) Service 作为到远程 JNDI 服务器的桥梁。
expander (扩散程序)	(n.)电子邮件传送系统的组成部分。它允许消息传送到地址列表。邮件扩散程序用于实现邮件列表。用户将消息发送到单个地址 (例如, users@example.com), 然后邮件扩散程序负责传送到列表中的邮箱。又称为邮件分发器。另参见 EXPN command (EXPN 命令) 。
expansion (扩散)	(n.)将发送地址为邮件列表的消息转换为足够满足邮件列表中每个成员的副本数的操作。适用于邮件列表的 MTA 处理。
expires header (到期标头)	(n.)远程服务器指定的返回文档的到期时间。
EXPN command (EXPN 命令)	(n.)用于扩展邮件列表的 SMTP 命令。在 RFC 821 中定义。

- expunge a message (清除邮件)** (v.)永久移除已从“收件箱”中删除的消息。另参见 [delete a message \(删除消息 \)](#)、[purge a message \(清除消息 \)](#)。
- extended simple mail transfer protocol (扩展的简单邮件传输协议)** (n.)一种 Internet 消息传输协议。ESMTP 将可选命令添加到 SMTP 命令集以获得增强功能，包括 ESMTP 服务器搜索远程站点所实现命令的功能。
- extensible markup language (可扩展标记语言)** 参见 [XML](#)。
- extensible style language (可扩展样式语言)** 参见 [XSL](#)。
- extensible style language transformation (可扩展样式语言变换)** 参见 [XSLT](#)。
- external entity (外部实体)** (n.)作为外部 [XML](#) 文件存在的实体，通过使用 [entity reference \(实体引用 \)](#) 将其包括在 XML 文档中。
- external subset (外部子集)** (n.)由对外部 DTD 文件的引用定义的那个 [DTD](#) 部分。
- extracting (提取)** (n.)查找文档中超文本链接的过程。每个被提取的链接将添加到 URL 池中进一步处理。
- extranet (外联网)** (n.)公司内联网向 Internet 的延伸，以允许客户、供应商和远程工作人员访问数据。

F

- facade (门面)** (n.)使用应用程序特定的有状态会话 Bean 管理各种 Enterprise JavaBeans™ 组件的位置。
- facility (工具)** (n.)在 Messaging Server 日志文件条目中, 表示生成日志条目的软件子系统(如“网络”或“帐户”)的名称。
- factory class (工厂类)** (n.)创建持久性管理器的类。另参见 [connection factory \(连接工厂\)](#)
- failover (故障转移)** (1)(n.)一种恢复过程。可以通过该过程透明地将会话、servlet 或有状态会话 bean 的状态从服务器崩溃中拯救出来。另参见 [persistence \(持久性\)](#)、[session failover \(会话故障转移\)](#)。
(2)(n.)计算机服务从一个系统向另一个系统的自动转移, 目的是提供冗余备份。
- family group administrator (系列组管理员)** (n.)拥有在系列组中添加和删除系列成员的管理权限的用户。此用户可以授予访问组中其他成员的系列组管理权限。
- fancy indexing (复杂索引)** (n.)提供比简单索引更多信息的索引方法。复杂索引按名称显示目录列表以及文件大小、上次修改日期和指示文件类型的图标。因此, 客户机加载复杂索引所需的时间可能比简单索引更长。
- fatal error (致命错误)** (n.)如果文档结构不正确或系统无法处理文档, 则 SAX 解析程序会出现致命错误。另参见 [warning \(警告\)](#)。
- federated identity (联合身份)** (n.)一个用户访问的所有服务提供者中的帐户信息(如个人数据、验证信息、消费习惯和历史记录、消费偏好等)的合并。该信息由用户管理, 并且可以在用户的同意下, 安全地与用户选择的提供者共享。
- federation cookie (联合 cookie)** (n.)联合 cookie 是由名为 fedCookie 的 Access Manager 实现的 cookie。根据主体的联合状态, 其值可为 yes 或 no。它不是 LAP 规范的已定义部分。
- federation termination (联合终止)** (n.)一个过程, 用户通过该过程取消在用户的身份认证提供者和联合服务提供者帐户间建立的关联关系。又称为取消联合。
- file cache (文件高速缓存)** (n.)文件高速缓存含有关于文件和静态文件内容的信息。默认情况下, 文件高速缓存处于启用状态。

- file extension (文件扩展名)** (n.)文件名称的最后部分，通常定义文件的类型。例如，在文件名 `index.html` 中，文件扩展名为 `html`。
- file transfer protocol (文件传输协议)** 参见 [FTP](#)。
- file type (文件类型)** (n.)给定文件的格式。例如，图形文件与文本文件具有不同的文件类型。文件类型通常由其文件扩展名标识。另参见 [file extension \(文件扩展名\)](#)。
- filter (过滤器)** (1) (n.)在搜索请求中，搜索范围内的条目要在搜索响应中返回所必须匹配的模式。过滤器还用于角色构建和访问控制定义中。
(2) (n.)一组定义特定资源类型的规则。站点定义使用这些过滤器定义 `robot` 应该接受或忽略的资源类型。
(3) (n.)可变换请求或响应的标题或内容（或二者）的对象。与 Web 组件不同，过滤器通常并不自己创建响应，而是修改或调整资源请求以及修改或调整来自资源的响应。过滤器不应对充当过滤器的 Web 资源具有信赖性，这样它才可由多种类型的 Web 资源组成。
- filter chain (过滤器链)** (n.)一种 [XSLT](#) 变换的串联，其中一个变换的输出变为下一个变换的输入。
- filtered role (过滤角色)** (n.)将角色分配给条目的方法。允许根据每个条目所含的属性，将条目分配给角色。这通过指定 LDAP 过滤器完成。符合过滤器要求的条目被视为拥有角色。
- filtering (过滤)** (n.)确定文档是否为某些站点的组成部分的过程，这些站点将纳入索引。
- finder method (finder 方法)** (n.)在 [home interface \(home 接口\)](#) 中定义的一种方法，使客户机可以在全局可用目录中查找一个 [entity bean \(实体 bean\)](#) 或一组 bean。
- firewall (防火墙)** (n.)保护组织内的联网计算机免受外部访问的网络配置，通常包括硬件和软件。防火墙一般用于保护物理建筑或组织站点内部的信息，如网络的电子邮件和数据文件等。
- flexible log format (灵活日志格式)** (n.)服务器用于将信息输入访问日志的格式。
- folder (文件夹)** (n.)已命名的消息集合。文件夹中可以含有其他文件夹。又称为邮箱。另参见 [personal folder \(个人文件夹\)](#)、[public folder \(公共文件夹\)](#)、[shared folder \(共享文件夹\)](#)、[INBOX](#)。
- form action handler (表单操作处理程序)** (n.)servlet 或应用程序逻辑中特别定义的方法。该方法基于表单上的命名按钮执行操作。
- form-based authentication (基于窗体的验证)** (n.)一种验证机制，其中 Web 容器提供特定于应用程序的窗体来实现登录。此验证形式使用 Base 64 编码，并且除非所有连接都在 SSL 上进行，否则可公开用户名和密码。
- FORTEZZA** (n.)美国政府机构使用的一种加密系统，用于管理敏感但未分类的信息。
- forwarding (转发)** 参见 [message forwarding \(消息转发\)](#)。
- foundation profile (基础配置文件)** (n.)一组 API 以及 CDC。它们共同提供面向下一代应用程序、消费电子产品和嵌入式设备的 J2ME™ 应用程序运行环境。

fragmentation (分段)	(n.)一种 MIME 功能, 可将较大的消息拆分成多个较小的消息。另参见 defragmentation (取消分段) 。
fresh start (全新启动)	(n.)从 robot 的起始点启动 robot。全新启动会删除 robot 的状态信息, 从而使 robot 从其初始状态开始其下一次运行。与重新启动相对。
FSMO role (FSMO 角色)	(Flexible Single-Master Operation role, 灵活单主操作角色) (n.)一种“活动目录”所使用的机制, 用于防止多主复制部署中发生更新冲突。即使为多主部署, 某些对象仍以单主模式更新, 这与 Windows NT 域中的主域控制器 (Primary Domain Controller, PDC) 的旧概念非常类似。“活动目录”部署中有五种 FSMO 角色, 但是只有 PDC 仿真器角色会影响适用于 Windows 的身份认证服务。因为密码更新只会立刻复制给具有 PDC 仿真器角色的“活动目录”域控制, 适用于 Windows 的身份认证服务使用该域控制器实现同步。否则与 Directory Server 之间的同步可能会延迟数分钟。
FTP	(file transfer protocol, 文件传输协议) (n.)一种 Internet 协议。该协议使得文件可以通过网络从一台计算机传送到另一台计算机。
fully qualified domain name (全限定域名)	(n.)系统的全名, 包含其主机名及其域名。例如: example.sun.com, 其中的 example 是 (服务器的) 主机名, sun.com 是域名。

G

- gateway (网关)** (n.)从一种本机格式转换成另一种本机格式的系统。示例如：往返于 X.400 和 RFC 822 之间的电子邮件网关。连接两个或更多电子邮件系统（特别是两个不同网络中的不同类型的邮件系统）并在其间传输消息的机器。有时，映射和转换会比较复杂。通常需要一个存储/转发方案，消息根据此方案从一个系统完全接收后，才经过适当转换，然后传输到下一个系统。
- general access (通用访问)** (n.)一种用户访问级别。授予此访问权限后，即表示所有通过验证的用户均可访问目录信息。
- general ACL (通用 ACL)** (n.)Java Enterprise System Directory Server 中的命名存取控制表，它将用户或组与一个或多个权限相关联。此列表可随意定义和访问，以记录任何权限设置。
- general entity (一般实体)** (n.)作为 XML 文档内容的一部分而被引用的实体，与参数实体截然不同，后者在 DTD 中引用。一般实体可以是解析实体或非解析实体。
- generation (生成)** (n.)robot 运行的一个阶段。在该阶段中，robot 为在枚举阶段中搜索到的每个资源生成一个资源描述。
- generic servlet (通用 servlet)** (n.)扩展 javax.servlet.GenericServlet 的 servlet。通用 servlet 独立于协议：它们本身不含对 HTTP 或任何其他传输协议的支持。另参见 [HTTP servlet](#)。
- GIF** (graphics interchange format, 图形交换格式) (n.)最初由 CompuServe 创建的一种跨平台图像格式。GIF 文件通常比其他图形文件类型（例如 TIFF）小很多。GIF 是最常用的交换格式之一。GIF 图像可直接在 UNIX、Microsoft Windows 和 Apple Macintosh 系统上显示。
- global database connection (全局数据库连接)** (n.)对多个组件可用的数据库连接。需要资源管理器。
- global transaction (全局事务)** (n.)由事务管理器管理和协调，并可跨多个数据库和进程的事务。事务管理器通常使用 XA 协议与数据库后端交互。另参见 [local transaction \(本地事务\)](#)。
- GMT** (Greenwich Mean Time, 格林尼治标准时间) (n.)英格兰格林尼治子午线的标准太阳时间，也是世界上所有其他时区参考的时间标准。GMT 不受“夏时制”的影响。

- granularity level (粒度级别)** (n.)将应用程序分成多个部分的方法。**高粒度级别**意味着将应用程序划分成许多更小的、定义更具体的 Enterprise JavaBeans™ 组件。**低粒度级别**意味着应用程序被划分成较少的部分，从而生成较大的程序。
- greeting form (问候表单)** (n.)通常在为用户创建帐户后向其发送的消息。此表单用于确认新帐户并验证其内容。
- group (组)** (1)(n.)按一般特性（如作业标题或由客户配置文件）进行分类的已验证用户集。组也与角色集相关，每个作为组成员的用户都将继承分配给该组的所有角色。组的两种类型分别为默认用户组和标准用户组。组成员资格通常由本地系统管理员维护。另参见 [user \(用户\)](#)、[role \(角色\)](#)。
- (2)(n.)以某个 [distinguished name \(标识名\)](#) 组织的若干个 LDAP 邮件条目。通常用作邮件列表，但也可用于将某些管理权限授予组内某些成员。另参见 [dynamic group \(动态组\)](#)、[static group \(静态组\)](#)。
- group folders (组文件夹)** (n.)这些文件夹包含共享文件夹和组文件夹。另参见 [public folder \(公共文件夹\)](#)、[shared folder \(共享文件夹\)](#)。
- group ID (组 ID)** (n.)Calendar Server 文件（如计数器和日志）的组。组 ID 存储于 `ics.conf` 文件的 `local.servergid` 参数中。又称为 `GID`。
- group scheduling engine (组调度引擎)** (n.)处理组调度的 Calendar Server 进程。此引擎使用户可与位于相同服务器或不同服务器上的其他日历用户一起安排事件。然后，其他用户可以修改、取消或回复该事件。
- GUI** (n.)graphical user interface（图形用户界面）。

H

HA	参见 high availability （高可用性）。
HA data service（HA 数据服务）	参见 data service （数据服务）。
HADB	参见 high availability database （高可用性数据库）。
HADB node（HADB 节点）	(n.)一组 HADB 进程、一个专用共享存储器区域及一个或更多个用于存储和更新会话数据的辅助存储设备。每个活动（数据存储）节点必须有一个镜像节点，因此节点都是成对出现的。此外，可以包含两个或更多个备用节点以使可用性达到最大。如果活动节点发生故障且无法在超时期间内恢复，备用节点会从镜像节点复制数据并转为活动状态。另参见 high availability database （高可用性数据库）
handle（句柄）	(n.)标识 enterprise bean （企业 bean）的对象。客户机可以使句柄序列化，以后再取消它的序列化，以获得对相应 bean 的引用。
hard restart（硬重启）	(n.)进程或服务的终止及其随后的重新启动。另参见 soft restart （软重启）。
hashdir	(n.)一种命令行实用程序，用于确定哪个目录含有某特定用户的消息存储。
HDML	（Handheld Device Markup Language，手持设备标记语言）(n.)Openwave 的用来为使用 Openwave 浏览器的移动设备编程的专用语言。
header（邮件头）	(n.)电子邮件消息中正文前面的部分。邮件头由字段名及值组成，字段名与值以冒号隔开。邮件头包含电子邮件程序以及想了解消息的用户所需的信息。例如，邮件头包括传送信息、内容摘要、跟踪和 MIME 信息。邮件头表明消息的接收者、发送者、发送时间及主题。邮件头必须按照 RFC 822 编写，以便电子邮件程序可以读懂。
header field（头字段）	(n.)邮件头中的命名项信息，如 "From:" 或 "To:"。又称为头行。
heartbeat（心跳）	(n.)在 Application Server 中，发送给群集中所有可用服务器的周期性消息。如果在指定的时间间隔和重试次数后缺少心跳，则可能触发 failover （故障转移）。
heuristic decision（启发式决策）	(n.)特定事务使用的事务模式。事务必须“提交”或“回滚”。

- high availability (高可用性)** (n.)在系统失败或出现处理故障时能够检测到服务中断, 并提供恢复机制。另外, 高可用性使备份系统在主系统出现故障时能够接管服务。又称为 HA。
- high availability database (高可用性数据库)** (HADB) (n.)一种高度可伸缩、高度可用的会话状态持久性基础结构。Application Server 使用 HADB 存储 HTTP 会话状态和有状态会话 bean 状态。另参见 [HADB node \(HADB 节点\)](#)、[active node \(活动节点\)](#)。
- home handle (主句柄)** (n.)一种对象, 可用于获得对 home 接口的引用。可以对主句柄进行序列化并将其写入稳定的存储器, 以及将其反序列化以获得引用。
- home interface (home 接口)** (n.)定义一类方法的接口, 此类方法使客户机能创建和删除 EJB 1.x 或 2.x [enterprise bean \(企业 bean\)](#)。会话 bean 的 home 接口定义 create 和 remove 方法, 而实体 bean 的 home 接口定义 create、finder 以及 remove 方法。另参见 [remote interface \(remote 接口\)](#)。
- home page (主页)** (n.)服务器上的一个文档, 充当服务器内容的目录或入口点。此文档的位置在服务器的配置文件中定义。
- hop (转发)** (n.)两台计算机之间的传输。
- horizontal scalability (水平可伸缩性)** (n.)Calendar Server 既可以在单个服务器上运行, 又可以作为一组进程, 使用多种可能配置选项分散在多个服务器上运行的能力。
- host (主机)** (n.)驻留有一个或多个服务器的计算机。
- hosted domain (托管域)** (n.)一种外包给 ISP 的电子邮件域。即, ISP 通过为某一组织运行和维护电子邮件服务为该公司提供电子邮件域托管功能。托管域与其他托管域共享同一 Java Enterprise System Messaging Server 主机。在早期基于 LDAP 的电子邮件系统中, 域由一个或多个电子邮件服务器主机支持。使用 Messaging Server, 单个服务器可以托管许多域。对于每一个托管域, 都有一个指向该域的用户和组容器的 LDAP 条目。又称为虚拟托管域或 [virtual domain \(虚拟域\)](#)。
- host-IP authentication (主机 IP 验证)** (n.)用于限制对 Java Enterprise System Administration Server 或 Web 站点上的文件和目录的访问的安全机制, 具体实现方法是使它们只对使用特定计算机的客户机可用。
- host name (主机名)** (n.)域内某一特定计算机的名称。主机名是 IP 主机名, 它可以是“简写”的主机名 (例如, mail) 或全限定主机名。全限定主机名由主机名和 [domain name \(域名\)](#) 组成。例如, mail.example.com 表示域 example.com 中的主机名 mail。在其域内主机名必须是唯一的。您的组织中可以有名为 mail 的多个计算机, 只要这些计算机驻留在不同的子域中, 例如, mail.corp.example.com 和 mail.field.example.com。主机名总是映射到特定的 IP 地址。另参见 [fully qualified domain name \(全限定域名\)](#)、[IP address \(IP 地址\)](#)。
- host-name hiding (主机名隐藏)** (n.)使用基于域的电子邮件地址的做法, 这些地址不包含特定内部主机的名称。
- HTML** (hypertext markup language, 超文本标记语言) (n.)Internet 上的超本文档的标记语言。HTML 能嵌入图像、声音、视频流、表单字段、对其他具有 URL 的对象的引用以及基本文本格式化。每一文本块都由表明文本性质的代码包围。
- HTML page (HTML 页面)** (n.)以 HTML 编码、专供在 Web 浏览器中显示的页面。

HTTP	(hypertext transfer protocol, 超文本传输协议) (n.)基于 Transmission Control Protocol/Internet Protocol (传输控制协议/Internet 协议) 的 Internet 协议, 它可从远程主机获取超文本对象。HTTP 消息包括从客户机到服务器的请求以及从服务器到客户机的响应。
HTTPD	(hypertext transfer protocol daemon, 超文本传输协议守护进程) (n.)HTTP 守护进程或服务的缩写, 它是一种使用 HTTP 协议供应信息的程序。
HTTP-NG	(hypertext transfer protocol-next generation, 下一代超文本传输协议) (n.)下一代超文本传输协议。
HTTPS	(hypertext transfer protocol secure, 安全超文本传输协议) (n.)使用 secure socket layer (安全套接字层) 协议实现的安全版本的 HTTP。
HTTP servlet	(n.)扩展 <code>javax.servlet.HttpServlet</code> 的 servlet。这些 servlet 具有对 HTTP 协议的内置支持。另参见 generic servlet (通用 servlet) 。
hub (集线器)	(n.)充当系统的单个接触点的主机。例如, 两个网络由防火墙分隔时, 防火墙计算机通常充当邮件集线器。
hypertext transfer protocol secure (超文本安全传输协议)	参见 HTTPS 。

iCalendar Message-Based Interoperability Protocol (基于 iCalendar 消息的互操作性协议)	(n.)此协议指定从 iCalendar Transport-Independent Interoperability Protocol (独立于 iCalendar 传输的互操作性协议) 到基于 Internet 电子邮件传输的绑定。此协议又称为 iMIP。iMIP 在 RFC 2447 中定义。
iCalendar Transport-Independent Interoperability Protocol (独立于 iCalendar 传输的互操作性协议)	(n.)一种基于 iCalendar 对象规范的 Internet 协议, 它提供不同日历系统之间的调度互操作性。此协议又称为 iTIP。iTIP 在 RFC 2446 中定义。
IDE	(integrated development environment, 集成开发环境) (n.)允许从单个图形用户界面创建、组合、部署和调试代码的软件。
IDENT	参见 Identification Protocol (标识协议)。
Identification Protocol (标识协议)	(n.)这种协议提供了确定远程进程的身份的方法, 适用于负责特定 TCP 连接的远端的远程进程。此协议又称为 IDENT。在 RFC 1413 中定义。
identity (身份)	(n.)用于明确区分某一位最终用户的一组信息。通过定义用户标识符和密码、电子邮件地址、个人偏好(例如音乐风格或选择性进/出市场决策)以及特定业务独有的其他信息(银行帐号或收货地址), 最终用户可以将自己与其他同样使用该服务的用户区分开来。
identity federation (身份联合)	(n.)用户选择将不同服务提供者帐户与身份认证提供者帐户联合起来时所发生的过程。用户保留在每个提供者处的个人帐户信息, 同时建立一个允许在提供者帐户之间交换验证信息的链接。也称为帐户联合。
identity provider (身份认证提供者)	(n.)专门提供验证服务的提供者。作为验证的管理服务, 身份认证提供者维护并管理身份验证信息。对于身份认证提供者提供的验证, 其所有联合服务提供者都认可。
identity service (身份认证服务)	(n.)身份认证服务是一种作用于资源的 Web 服务, 可以检索、更新与“主体”(身份)有关的数据属性或者对这些属性执行某些操作。身份认证服务的示例有公司电话簿或日历服务。

IDL	(interface definition language, 接口定义语言) (n.)用于定义与远程 CORBA 对象的接口的语言。这些接口独立于操作系统和编程语言。该语言描述远程过程调用 (remote procedure call, RPC) 的功能接口, 以便编译程序能够生成在计算机之间对参数进行编组的代理和桩模块代码。
idle state (空闲状态)	(n.)一种状态。在此状态下, robot 仍然在运行, 但是已经处理了其 URL 池中的所有 URL。此时, robot 仍然可以对状态请求做出响应。
iHTML	(i-mode hypertext markup language, i-mode 超文本标记语言) (n.)NTT DoCoMo 的日语 i-mode 服务使用的语言。
IOP	(Internet Inter-ORB Protocol, Internet ORB 互联协议) (n.)基于 IOP 上的远程方法调用 (Remote Method Invocation, RMI) 以及公共对象请求代理体系结构 (Common Object Request Broker Architecture, CORBA) 均使用的一种传输层协议。用于 CORBA 对象请求代理之间的通信。
IOP cluster (IOP 群集)	(n.)为获得 RMI / IOP 请求的高可用性而配置的 IOP 群集。
IOP endpoint (IOP 端点)	(n.)为 IOP 群集配置的 IOP 侦听器, 其用途是实现 RMI / IOP 请求的高可用性。
IOP listener (IOP 侦听器)	(n.)侦听指定端口并接受来自基于 CORBA 的客户机应用程序的外来连接的侦听套接口。
imagemap	(1) (n.)一种激活图像某些区域的进程, 使用户可通过用鼠标单击图像的不同区域来导航并获取信息。 (2) (n.)一种 CGI 程序, 用于执行其他 HTTPD 实现中的 imagemap 功能。
IMAP4	(Internet Message Access Protocol Version 4, Internet 消息访问协议版本 4) (n.)一种标准协议, 使用户可从主消息传送系统断开连接而仍然能够处理其邮件。 IMAP 规范允许对这些断开连接的用户进行管理控制, 并且在他们重新连接到消息传送系统时同步他们的消息存储。
iMIP	参见 iCalendar Transport-Independent Interoperability Protocol (独立于 iCalendar 传输的互操作性协议)。
immediate subordinate (直接下属)	(n.)在 DIT 中, 如果某个条目的 distinguished name (标识名) 是通过将其 relative distinguished name (相对标识名) 附加到另一条目 (父条目) 的标识名得到的, 则此条目是该父条目的直接下属。
immediate superior (直接上级)	(n.)在 DIT 中, 如果某个条目的 distinguished name (标识名) 附加另一条目 (子条目) 的 relative distinguished name (相对标识名) 后即构成该子条目的标识名, 则此条目是该子条目的直接上级。
impersonation (身份模拟)	(n.)一种操作, 通过该操作一个对象可以使用另一个对象的身份和权限, 而不会存在任何限制以及任何使模拟方呼叫接收者看到已发生委托的迹象。身份模拟是一种简单的 delegation (委托)。
import agent (导入代理)	(n.) importing (导入) 期间使用的进程。

importing (导入)	(n.)将新的或更新的资源描述从另一数据库中引入“搜索引擎”的过程。
imsadmin commands (imsadmin 命令)	(n.)用于管理域管理员、用户和组的一组命令行实用程序。
imsimta commands (imsimta 命令)	(n.)用于执行各种 MTA 维护、测试和管理任务的一组命令行实用程序。
inactive boot environment (非活动引导环境)	(n.)当前未引导或未被指定为在下次重新引导时激活的环境。另参见 active boot environment (活动引导环境) 。
INBOX	(n.)用户默认邮箱的保留名称。用于邮件传送。 INBOX 是唯一的区分大小写的文件夹名，即 INBOX 、 Inbox 和 inbox 都是用户默认邮箱的有效名称。
index (索引)	(n.)资源或文档的集中化并可搜索的数据库。又称为目录。
indexing (建立索引)	(n.)提供集中化、可搜索的资源数据库的过程。又称为编目。
index key (索引键)	(n.)目录使用的每个索引由一个索引键表和相应的条目 ID 列表组成。
indirect CoS (间接 CoS)	(n.)用一个目标条目属性的值来标识模板条目。
initialization parameter (初始化参数)	(n.)一种 parameter (参数) ，用于初始化与 servlet 相关的上下文。
inittab file (inittab 文件)	(n.) (仅限 UNIX) 一种文件，用于列出由于任何原因停止后需要重新启动的程序。该文件确保程序连续运行。由于其位置的关系，该文件又称为 /etc/inittab 。并非所有 UNIX 系统上都有此文件。
installation directory (安装目录)	(n.)安装服务器二进制 (可执行) 文件的目录。对于 Messaging Server ，安装目录是 server root (服务器根目录) 的子目录： server-root/bin/msg/ 。另参见 instance directory (实例目录) 。
installation path (安装路径)	(n.)安装 Directory Server Enterprise Edition 软件的完整路径。可在首次安装软件时选择安装路径。
instance directory (实例目录)	(n.)包含定义服务器特定实例的文件的目录。对于 Messaging Server ，实例目录是 server root (服务器根目录) 的子目录： server-root/msg-instance/ ，其中 instance 是安装时指定的服务器名称。对于 Application Server ，实例目录是 domain directory (域目录) 的子目录。另参见 installation directory (安装目录) 、 server instance (服务器实例) 。
instance path (实例路径)	(n.) Directory Server 或 Directory Proxy Server 服务器实例数据所在位置的完整路径。在创建服务器实例时选择实例路径。
Instant Messaging Client (Instant Messaging 客户机)	(n.)使用户能够发送和接收即时消息和警报的客户机。

- Instant Messaging multiplexor (Instant Messaging 多路复用器)** (n.)一个客户机连接管理器。它使大量并发客户机连接只需使用少量与后端 Instant Messaging 服务器的连接，从而提高 Instant Messaging Server 的可伸缩性。Instant Messaging 客户机连接到此多路复用器，而不是连接到 Instant Messaging 服务器本身。安装在防火墙的公共侧时，此多路复用器保护用户数据库不受入侵者的侵犯，将 Instant Messaging Server 保护在防火墙之后。
- Instant Messaging Server** (1) (n.)指 Java Enterprise System Messaging Server 产品自身，包括所有组件（服务器、多路复用器和 Java Enterprise System Instant Messaging Server）。
- (2) (n.)该产品的后端服务器进程，用于处理来自 Instant Messaging（通过 Instant Messaging Server 多路复用器）的外来命令。验证 Instant Messaging 用户时，Instant Messaging Server 还与 LDAP 服务器通信。另参见 [Instant Messaging multiplexor \(Instant Messaging 多路复用器\)](#)
- intelligent agent (智能代理)** (n.)服务器中代表用户执行各种请求（如 HTTP、NNTP、SMTP 和 FTP 请求）的对象。在某种意义上，智能代理充当服务器的客户机，发出服务器完成的请求。
- international index (国际索引)** (n.)一种搜索索引。加速搜索 DIT（其中的属性有语言标记）中的信息。
- Internet Message Access Protocol Version 4 (Internet 消息访问协议版本 4)** 参见 [IMAP4](#)。
- Internet Protocol (Internet 协议)** 参见 [IP](#)。
- intranet (内联网)** (n.)公司或组织内多个 Transmission Control Protocol/Internet Protocol（传输控制协议/Internet 协议）网络组成的网络。对于分布在公司 LAN 上的内部应用程序，内联网使公司能够使用相同类型的万维网服务器和客户机软件。内联网上与 Internet 通信的敏感信息通常用防火墙加以保护。另参见 [firewall \(防火墙\)](#)、[extranet \(外联网\)](#)。
- invalid user (无效用户)** (n.)消息处理过程中出现的错误情况。出现此错误情况时，消息存储向 MTA 发送通信，然后删除它拥有的消息副本。MTA 将消息发回给发件人并删除其消息副本。
- IP** (Internet Protocol, Internet 协议) (n.)传输控制协议/Internet 协议族中用于在世界范围内链接网络的协议。由美国国防部开发，并在 Internet 中采用。此协议族的突出点是 IP 协议。
- IP address (IP 地址)** (n.)由点分隔的一组数字（如 192.168.255.255），指定内联网或 Internet 上某台计算机的实际位置。是分配给使用 Transmission Control Protocol/Internet Protocol（传输控制协议/Internet 协议）的主机的 32 位地址。
- ISDN** (n.)Integrated Services Digital Network（综合业务数字网）。
- ISINDEX** (n.)在客户机中启用搜索的 HTML 标记。文档可以使用网络导航器的功能来接受搜索字符串，然后将其发送到服务器以访问可搜索索引，而无需使用表单。要使用 ISINDEX HTML 标记，必须创建查询处理程序。
- ISMAP** (n.)HTML 文档中所用的 IMG SRC 标记的扩展，用于告知服务器指定的图像为 [imagemap](#)。

ISO 3166	(n.)由国际标准化组织 (International Organization for Standardization, ISO) 所维护的国家/地区代码的国际标准。
ISO 8601	(n.)一项“国际标准化组织”标准，用于指定日期与时间的数字表示法。Calendar Server 使用 ISO 8601 标准表示法表示日期、时间和持续时间字符串。
isolation level (隔离级别)	参见 transaction isolation level (事务隔离层) 。
issued certificate (已颁发证书)	(n.)由 certificate authority (证书授权机构) 颁发的证书。另参见 self-generated certificate (自生成证书) 。
ISV	(n.)Independent software vendor (独立软件供应商)。
iTIP	参见 iCalendar Transport-Independent Interoperability Protocol (独立于 iCalendar 传输的互操作性协议) 。

J

J2EE™ application (J2EE 应用程序)	(n.) J2EE platform (J2EE 平台) 功能的任意可部署单元。可以是单个 J2EE 模块, 或者是与 J2EE 应用程序部署描述符一起打包到 EAR 文件中的一组模块。J2EE 应用程序通常被设计为分布于多个计算层。
J2EE component (J2EE 组件)	(n.)一种受容器支持的自包含功能性软件单元, 在部署时可对其进行配置。包括 web application (Web 应用程序) 、 session bean (会话 bean) 、 entity bean (实体 bean) 、 message-driven bean (消息驱动 bean) 、 application client (应用程序客户端) 或 connector (连接器) 。这些 J2EE 组件以 Java™ 编程语言编写, 并且编译方式与该语言其他程序相同。另参见 component (组件) 。
J2EE module (J2EE 模块)	(n.)一种软件单元, 包含一个或多个具有相同容器类型以及该类型的一个部署描述符的 J2EE 组件。模块可以作为独立单元来部署, 也可以集成到 J2EE 应用程序中。另参见 life-cycle module (生命周期模块) 、 module (模块) 。
J2EE platform (J2EE 平台)	(Java 2 Platform, Enterprise Edition) (n.) 用于开发和部署多层、基于 Web 的企业应用程序的环境。J2EE 平台由提供这些应用程序开发所需功能的一组服务、 API 和协议组成。
J2EE product (J2EE 产品)	(n.)符合 J2EE 平台规范的实现。
J2EE product provider (J2EE 产品提供者)	(n.)提供 J2EE 产品的供应商。
J2EE server (J2EE 服务器)	(n.)J2EE 产品的运行时部分。J2EE 服务器提供 EJB 或 Web 容器, 或两者同时提供。另参见 container (容器) 。
J2ME™ platform (J2ME 平台)	(Java 2 Platform, Micro Edition) (n.)高度优化的 Java 运行时环境, 其面向广泛的消费类产品, 包括寻呼机、便携式电话、可视电话、数字机顶盒以及车载导航系统。
J2SE™ platform (J2SE 平台)	(Java 2 Platform, Standard Edition) (n.)核心 Java 技术平台。
JAF	(JavaBeans™ Activation Framework , JavaBeans 激活框架) (n.)将 MIME 数据类型支持集成到 Java 平台中。另参见 MIME data type (MIME 数据类型) 。

JAR file contract (JAR 文件订约)	(n.)Java 归档文件订约, 指定企业 bean 软件包中必须包含哪些信息。
JATO	(n.)一种库, 用于相互转换以 Java 编程语言编写的代码和以 XML 编写的代码。又称为 Sun Java System Web Application Framework 或 Application Framework。JATO 适合于企业 Web 应用程序开发。JATO 结合了多种概念, 如显示字段、应用程序事件、组件分层结构以及以页面为中心的开发方法。
Java 2 Platform, Enterprise Edition	参见 J2EE platform (J2EE 平台) 。
Java 2 Platform, Micro Edition	参见 J2ME platform (J2ME 平台) 。
Java 2 Platform, Standard Edition	参见 J2SE platform (J2SE 平台) 。
JavaBean™ namespace (JavaBean 名称空间)	(n.)一种标准, 允许为由某一软件包定义的元素名称集指定唯一的标签。使用该软件包的文档可以加入任何其他文件, 而不会造成元素名称间的冲突。在软件包中定义的元素将被唯一地标识, 以便解析程序可以根据您的软件包而不是其他软件包确定何时解释一个元素。
JavaBeans Activation Framework (JavaBeans 激活框架)	参见 JAF 。
JavaBeans component (JavaBeans 组件)	(n.)可以由工具进行处理并可编写到应用程序中的 Java 类。JavaBeans 组件必须遵守特定属性和事件接口约定。
JavaBeans component architecture (JavaBeans 组件体系结构)	(n.)一种可移植的、独立于平台的可重用组件模型。
Java Enterprise System	(n.)由单个 Sun 软件产品集成的软件系统, 它支持分布式企业应用程序。
Java ES	参见 Java Enterprise System 。
Java ES shared component (Java ES 共享组件)	参见 shared component (共享组件) 。
JavaMail™ (API, extension) (JavaMail (API , 扩展))	(n.)用于发送和接收电子邮件的 API。应用程序代码使用 Java Naming and Directory Interface™ (JNDI) 服务查找采用 JNDI 名称的 JavaMail 会话资源对象。
JavaScript™ programming language (JavaScript 编程语言)	(n.)一种简洁的、基于对象的脚本语言, 用于开发客户机和服务器 Internet 应用程序。

JavaServer Faces™ conversion model (JavaServer Faces 转换模型)	(n.)一种转换机制，用于相互转换由 JavaServer Faces UI 组件生成的、基于字符串的标记和服务器端 Java 对象。
JavaServer Faces event and listener model (JavaServer Faces 事件和侦听器模型)	(n.)一种用于确定如何处理由 JavaServer Faces UI 组件产生的事件的机制。该模型基于 JavaBeans 组件事件和侦听器模型。
JavaServer Faces expression language (JavaServer Faces 表达式语言)	(n.)由 JavaServer Faces UI 组件标记属性使用的简单表达式语言，用于将相关组件绑定到 bean 属性上或将相关组件的值绑定到方法或外部数据源（如 bean 属性）上。与 JSP expression language (JSP 表达式语言) 表达式不同，JavaServer Faces EL 表达式由 JavaServer Faces 实现而非 Web 容器来计算。
JavaServer Faces navigation model (JavaServer Faces 导航模型)	(n.)一种用于定义 JavaServer Faces 应用程序中页面显示顺序的机制。
JavaServer Faces UI component (JavaServer Faces UI 组件)	(n.)一种用户界面控件，用于将数据输出到客户机或允许用户将数据输入 JavaServer Faces 应用程序。
JavaServer Faces UI component class (JavaServer Faces UI 组 件类)	(n.)定义 JavaServer Faces UI 组件行为和属性的 JavaServer Faces 类。
JavaServer Faces technology (JavaServer Faces 技术)	(n.)一种框架，用于为以 Java 编程语言编写的 Web 应用程序建立服务器端用户界面。
JavaServer Faces validation model (JavaServer Faces 验证模型)	(n.)一种机制，用于验证用户输入 JavaServer Faces UI 组件的数据。
JavaServer Pages™ technology (JavaServer Pages 技术)	参见 JSP technology (JSP 技术) 。
Java Web Start software (Java Web Start 软件)	(n.)一种 Web 应用程序启动程序。使用 Java Web Start 软件，单击 Web 链接即可启动应用程序。如果应用程序不在计算机上，Java Web Start 会自动下载该应用程序并将它缓存到计算机中。应用程序下载到其缓存区后，即可从桌面图标或浏览器链接启动。无论使用哪种方法启动应用程序，始终都会显示应用程序的最新版本。
JAXM	(Java API for XML Messaging, 用于 XML 消息传送的 Java API) (n.)一种使用 SOAP 标准的 Java API，使应用程序能够发送和接收面向文档的 XML 消息。这些消息可以有附件，也可以没有附件。

- JAXP** (Java API for XML Processing, 用于 XML 处理的 Java API) (n.) 用于处理 XML 文档的 API。JAXP 利用解析程序标准 SAX 和 DOM, 以便选择将数据解析为事件流或生成其树状结构的表示。JAXP 支持 XSLT 标准, 从而让您能够控制数据的表示方式并能将数据转换为其他 XML 文档或其他格式 (如 HTML)。JAXP 提供对名称空间的支持, 从而允许使用在其他情况下可能会产生命名冲突的模式。
- JAXR** (Java API for XML Registries, 用于 XML 注册表的 Java API) (n.) 一种统一的标准 Java API, 用于访问不同种类的 XML 注册表。它使用户能够构建、部署和搜索 Web 服务。另参见 registry (注册表)。
- JAXR client (JAXR 客户机)** (n.) 一种客户机程序, 它使用 JAXR API 通过 JAXR 提供者访问业务注册表。
- JAXR provider (JAXR 提供者)** (n.) JAXR API 的一种实现, 提供对特定注册表提供者的访问或对基于通用规范的注册表提供者类的访问。
- JAX-RPC** (Java API for XML-based RPC, 用于基于 XML 的 RPC 的 Java API) (n.) 一种 Java API, 使开发者能够根据基于 XML 的 RPC 协议, 构建可交互使用的 Web 应用程序和 Web 服务。
- JDBC™ connection pool (JDBC 连接池)** (n.) 将用于指定数据库连接的 JDBC 数据源属性与连接池属性结合在一起的池。
- JDBC resource (JDBC 资源)** (n.) 一种资源, 用于通过现有 JDBC 连接池, 将应用服务器中运行的应用程序连接到数据库。由 Java Naming and Directory Interface (JNDI) 名称 (供应用程序使用) 和一个现有 JDBC 连接池的名称组成。
- JDBC technology (JDBC 技术)** (Java DataBase Connectivity software, Java 数据库连接软件) (n.) 一组标准化的类和接口, 使开发者能够创建数据感知组件。JDBC API 以一种独立于平台和供应商的方式, 实现了连接到数据源以及与数据源交互的方法。JDBC 技术为基于 SQL 的数据库访问提供了调用级 API。
- JHTML** (J-Sky hypertext markup language, J-Sky 超文本标记语言) Vodafone 专有的语言, 用于为日语 J-Sky 设备编程。
- JMS** (Java Message Service, Java 消息服务) (n.) 定义 Java 客户机如何访问消息服务设备的一套标准接口和语义。这些接口为使用 Java 编程语言编写的程序提供了一种创建、发送、接收和读取消息的标准方式。
- JMS administered object (JMS 管理对象)** (Java Message Service administered object, Java 消息服务管理对象) (n.) 由管理员创建的预配置 Java 消息服务对象 (JMS connection factory (JMS 连接工厂) 或 JMS destination (JMS 目的地)), 供一个或多个 JMS 客户端使用。使用受管理对象可以使 JMS 客户端从提供者的所有权中脱离出来, 从而使客户机独立于提供者。这些对象由管理员置入一个 Java Naming and Directory Interface (JNDI) 名称空间中, 并由 JMS 客户端使用 JNDI 查找进行访问。
- JMS API** (Java Message Service API, Java 消息服务 API) (n.) 定义 JMS 客户端如何访问 JMS 消息服务设备的一套标准接口和语义。这些接口为使用 Java 编程语言编写的程序提供了一种创建、发送、接收和读取消息的标准方式。
- JMS application (JMS 应用程序)** (Java Message Service application, Java 消息服务应用程序) (n.) 一个或多个进行消息交换的 JMS 客户端。

JMS client (JMS 客户端)	(Java Message Service client, Java 消息服务客户机) (n.)一种应用程序或软件组件, 使用 JMS 消息服务与其他 JMS 客户端进行交互, 以交换消息。
JMS connection factory (JMS 连接工厂)	(Java Message Service connection factory, Java 消息服务连接工厂) (n.)由 Java 消息服务管理的对象, JMS 客户端使用该对象创建与 JMS 消息服务的连接。
JMS destination (JMS 目的地)	(Java Message Service destination, Java 消息服务目的地) (n.)JMS 消息服务中的物理目的地, 生成的消息被传送到此目的地, 以进行路由选择和继续传送给使用方。此物理目的地由一个 JMS 管理对象标识和封装, JMS 客户端使用该对象指定收到的消息和外出的消息的目的地。
JMS messages (JMS 消息)	(Java Message Service messages, Java 消息服务消息) (n.)由 Java 消息服务客户机使用的异步请求、报告或事件。消息包含消息头 (可以向其中添加附加字段) 和正文。消息头指定标准字段和可选属性。消息正文含有要发送的数据。
JMS provider (JMS 提供者)	(Java Message Service provider, Java 消息服务提供者) (n.)一种为消息传送系统提供 JMS 接口, 并添加完整产品所需的管理和控制功能的产品。
JMS service (JMS 服务)	(Java Message Service service, Java 消息服务的服务) (n.)为 Java 消息服务消息传送系统提供传送服务 (包括与 JMS 客户端的连接、消息路由选择和传送、持久性、安全性和日志记录) 的软件。消息服务维护的物理目的地接收 JMS 客户端发送的消息, 而该目的地的消息被发送至使用该消息的客户端。
JMS session (JMS 会话)	(Java Message Service session, Java 消息服务会话) (n.)用于发送和接收 JMS 消息的单线程上下文。JMS 会话可以是非事务性的、经本地事务处理的或者可以参与分布式事务。
JNDI extension (JNDI 扩展)	(Java Naming and Directory Interface extension, Java Naming and Directory Interface 扩展) (n.)Java 平台的一种标准扩展, 它为支持 Java 技术的应用程序提供使用企业中多个命名和目录服务的统一接口。作为 Java Enterprise API 集的组成部分, JNDI 可使各种不同的企业命名和目录服务相互连接。
JNDI name (JNDI 名称)	(Java Naming and Directory Interface name, Java Naming and Directory Interface 名称) (n.)用于访问已在 JNDI 命名服务中注册的资源的名称。
job controller (作业控制器)	(n.)负责在接到各种其他 MTA 组件请求时调度和执行任务的 MTA 组件。
join rule (联接规则)	(n.)一种规则, 用于指定 Directory Proxy Server 从数据视图中的条目与主数据视图中的条目的链接方式, 或者一个 SQL 表中的条目与其他 SQL 表中的条目的链接方式。
jspc utility (jspc 实用程序)	(n.)用于采用 JSP technology (JSP 技术) 创建的页面的编译程序。该实用程序检查所有 JSP 页是否符合 JSP 规范。
JSP™ action (JSP 操作)	(n.)一种 JSP 元素, 可以对隐含对象和其他服务器端对象进行操作, 或者可以定义新的脚本变量。操作遵守元素的 XML 语法, 有起始标记、正文和结束标记; 如果正文为空, 它也可以使用空标记语法。标记必须使用前缀。操作分为标准和自定义操作。
JSP container (JSP 容器)	(n.)一种 container (容器) , 其提供与 servlet 容器以及将 JSP 页解释并处理成 servlet 的引擎相同的服务。

JSP container, distributed (可分布的 JSP 容器)	(n.)一种 JSP 容器，可以运行被标记为可分布的 Web 应用程序，并且散布于可能运行在不同主机上的多个 Java 虚拟机上。
JSP custom action (JSP 自定义操作)	(n.)一种用户定义的操作，由标记库描述符以可移植方式描述并由 <code>taglib</code> 指令导入 JSP 页。自定义操作用于封装编写 JSP 页过程中的递归任务。
JSP custom tag (JSP 自定义标记)	(n.)一种引用 JSP 自定义操作的 tag (标记) 。
JSP declaration (JSP 声明)	(n.)一种 JSP 脚本元素，用在 JSP 页中声明方法、变量或两者。
JSP directive (JSP 指令)	(n.)一种 JSP 元素，向 JSP 容器提供指令并在转换时解释。
JSP document (JSP 文档)	(n.)以 XML 语法编写并受到 XML 文档约束的 JSP 页。
JSP element (JSP 元素)	(n.)由 JSP 转换程序识别的 JSP 页部分。元素可以是指令、操作或脚本元素。
JSP expression (JSP 表达式)	(n.)一种脚本元素，包含一个被计算和转换为 字符串 并放入隐含 <code>out</code> 对象的有效脚本语言表达式。
JSP expression language (JSP 表达式语言)	(n.)一种语言，用于编写表达式来访问 JavaBeans 组件属性。EL 表达式可用于静态文本以及任何可以接受表达式的标准或自定义标记属性。
JSP page (JSP 页)	(n.)一种基于文本的文档，其包含静态文本以及描述如何处理请求以创建响应的 JSP 元素。JSP 页转换为 <code>servlet</code> ，并像 <code>servlet</code> 一样处理请求。
JSP scripting element (JSP 脚本元素)	(n.)一种 JSP 声明、scriptlet 或表达式，其语法由 JSP 规范定义，其内容根据 JSP 页中所用的脚本语言来编写。JSP 规范描述了语言页面属性为 "java" 的情况下的语法和语义。
JSP scriptlet	(n.)一种 JSP 脚本元素，其包含在 JSP 页所用的脚本语言中有效的任意代码片段。JSP 规范描述了在语言页面属性为 "java" 的情况下什么是有效的 scriptlet。
JSP standard action (JSP 标准操作)	(n.)在 JSP 规范中定义的一种操作，并总是对 JSP 页可用。
JSP tag file (JSP 标记文件)	(n.)一种源文件，其包含 JSP 代码的可重用片段，该片段在 JSP 转换为 <code>servlet</code> 时将转换为标记管理器。
JSP tag handler (JSP 标记管理器)	(n.)一种 Java 编程语言对象，可实现自定义标记的行为。
JSP tag library (JSP 标记库)	(n.)使用标记库描述符和 Java 类来描述的自定义标记集合。另参见 JSTL 。
JSP™ technology (JSP 技术)	(1)(n.)一种可扩展的 Web 技术，它使用静态数据、JSP 元素以及服务器端 Java 对象来为客户机生成动态内容。通常，静态数据为 HTML 或 XML 元素，并且在多数情况下，客户机为 Web 浏览器。使用 JSP 技术创建的页面将标准浏览器页面的布局功能和编程语言的强大功能结合在一起。

(2)(n.)实现了所有 JSP 技术元功能（包括实例化、初始化、解构、从其他组件访问以及配置管理）的扩展。在 Web 服务器而非 Web 浏览器上运行的可再度使用的 Java 应用程序。

- JSS** 参见 [Network Security Services for Java, JSS](#)（面向 Java 的网络安全服务）。
- JSSE**（Java Secure Socket Extension, Java 安全套接字扩展）(n.)一组启用安全 Internet 通信的软件包。
- JSTL**（JavaServer Pages 标准标记库）(n.)一种标记库，用于封装对大量 JSP 应用程序通用的核心功能。JSTL 支持常见的结构化任务，如迭代和条件语句、用于处理 XML 文档的标记、国际化和特定于语言环境的格式化标记、SQL 标记以及函数。
- JTA**（Java transaction API, Java 事务 API）(n.)一种 API，它允许应用程序和 J2EE 服务器访问事务。
- JTS**（Java transaction service, Java 事务服务）(n.)指定事务管理器的实现，该事务管理器支持 JTA 并以低于 API 的级别实现符合 Object Management Group Object Transaction Service 1.1 规范的 Java 映射。

K

- key database (密钥数据库)** (n.)一种文件，其中包含一个或多个服务器证书的一个或多个密钥对。也称为密钥文件。
- key-pair file (密钥对文件)** 参见 [trust database \(信任数据库\)](#)。
- keystore (密钥库)** (n.)一个文件，其中包含用于验证的密钥和证书。
- knowledge information (知识信息)** (n.)目录服务基础结构信息的组成部分。目录服务器使用知识信息将对信息的请求传递给其他服务器。
- knowledge reference (知识引用)** (n.)指向存储在不同数据库中的目录信息的指针。

L

last-modified header (last-modified 标头)	(n.)在来自服务器的 HTTP 响应中返回的文档文件的最后修改时间。
LDAP	(Lightweight Directory Access Protocol, 轻量目录访问协议) (n.)专用于通过 TCP/IP 并跨多个平台运行的目录服务协议。X.500 目录访问协议 (Directory Access Protocol, DAP) 的简化形式, 允许对信息 (包括跨多个 Sun Java System 服务器的用户概要文件、分发列表以及配置数据) 的存储、检索和分发进行单点管理。Directory Server 使用 LDAP 协议。
LDAP database (LDAP 数据库)	(n.)一种数据库, 用于存储在验证中使用的用户和组列表。
LDAP data interchange format (LDAP 数据交换格式)	参见 LDIF 。
LDAP filter (LDAP 过滤器)	(n.)根据存在的特定属性或属性值指定条目集的方法。
LDAP referrals (LDAP 引用)	(n.)包含指向另一个 LDAP 条目的符号链接 (引用) 的 LDAP 条目。LDAP 引用由一个 LDAP 主机和一个标识名组成。LDAP 引用通常用于引用现有 LDAP 数据, 从而不必复制该数据。LDAP 引用还用于当程序所依赖的特定条目已经移动时, 保持该程序的兼容性。
LDAP search string (LDAP 搜索字符串)	(n.)含有可替换参数的字符串, 其中的参数定义用于目录搜索的属性。例如, LDAP 搜索字符串 “uid=%s” 表示基于用户 ID 属性进行搜索。
LDAP server (LDAP 服务器)	(n.)一种软件服务器, 可维护 LDAP 目录并为目录查询提供服务。“Sun 目录服务”和“Netscape 目录服务”是“LDAP 服务器”的实现。
LDAP server failover (LDAP 服务器故障转移)	(n.)LDAP 服务器的一种备份功能。如果一台 LDAP 服务器发生故障, 系统可以切换到另一台 LDAP 服务器。
LDAP URL	(n.)一种 URL, 它提供使用 DNS 来查找目录服务器, 然后通过 LDAP 完成查询的方法。ldap://ldap.example.com 是一个 LDAP URL 范例。
LDAPv3	(n.) LDAPv3 协议版本 3。

LDBM	(n.)LDAP database manager (LDAP 数据库管理器)。
LDBM database (LDBM 数据库)	(n.)一种基于磁盘的高性能数据库, 该数据库由包含 Directory Server 中所有数据的一组大文件组成。
LDIF	(LDAP Data Interchange Format, LDAP 数据交换格式) (n.)该格式使用 <i>type:value</i> 对并以文本形式表示 Directory Server 条目。
leaf entry (叶条目)	(n.)下面没有其他条目的条目。叶条目不能是目录树中的分支点。
Legato NetWorker® software (Legato NetWorker 软件)	(n.)由 Legato Systems, Inc. 分发的第三方备份实用程序。
level (级别)	(n.)日志记录冗余程度的说明, 表示日志文件中所记录的事件类型的相对数量。例如, “紧急”(SEVERE)级别将记录很少事件。而“信息”(INFO)级别将记录很多事件。
Liberty-enabled client (启用 Liberty 的客户机)	(n.)启用 Liberty 的客户机是指具有(或知道如何获得)有关身份认证提供者信息的客户机, 主体将使用该信息对服务提供者进行验证。
Liberty-enabled proxy (启用 Liberty 的代理)	(n.)启用 Liberty 的代理是指模拟启用 Liberty 的客户机的 HTTP 代理。
life cycle (生命周期)	(1) (n.)J2EE 组件存在的框架事件。每种类型的组件都具有标记其过渡到某些状态(在这些状态下, 该组件具有不同的可用性)的定义事件。例如, 创建 servlet 后, 在客户机(或其他需要 servlet 功能的 servlet)调用 servlet 的 <code>service</code> 方法前, 由 servlet 容器调用 servlet 的 <code>init</code> 方法。调用 <code>init</code> 方法后, 它便具有数据并准备用于预定用途。servlet 容器在 servlet 生命周期结束前调用其 <code>destroy</code> 方法, 这样就可以完成与结束相关的处理并释放资源。本例中的 <code>init</code> 和 <code>destroy</code> 方法属于 callback method (回调方法) 。类似的注意事项适用于所有 J2EE 组件类型的生命周期: 企业 bean、Web 组件(servlet 或 JSP 页)、applet 以及应用程序客户端。 (2) (n.)一组阶段, 在这些阶段会接收到 JavaServer Faces 页面请求、对表示页面的 UI 组件树进行处理以及产生响应。 (3) (n.)服务器运行时(包括从启动到关机)的框架事件。
life-cycle event (生命周期事件)	(n.)服务器生命周期的一个阶段, 如启动或关机。
life-cycle module (生命周期模块)	(n.)一个模块, 可侦听并执行其任务以响应服务器生命周期中的事件。
Lightweight Directory Access Protocol (轻量目录访问协议)	参见 LDAP 。
listener (侦听器)	(n.)在发布对象中注册的类, 说明事件发生时执行的操作。
listen port (侦听端口)	(n.)服务器用于与客户机和其他服务器通信的端口。

listen socket (侦听套接字)	(n.) port (端口) 号和 IP address (IP 地址) 的组合。服务器和客户机通过侦听套接字连接。
LMTP	(Local Mail Transfer Protocol, 本地邮件传输协议) (n.)与 SMTP 类似, 但不需要对邮件传送队列进行管理。另外, LMTP 为消息的每个收件人都提供一个状态代码, 而 SMTP 仅为消息提供一个状态代码。在 RFC 2033 中定义。
load balancer (负载均衡器)	(n.)一种软件, 可控制与多台网关机器的连接, 使每个可用系统上的负载基本相等。
load balancing (负载均衡)	(n.)在群集中的多个节点上分配应用程序负载从而使客户机请求及时得到服务的过程。仅适用于可伸缩服务。
local database connection (本地数据库连接)	(n.)本地连接中的事务上下文对当前进程以及当前数据源是本地关系, 而不是跨进程或跨数据源分布。
locale (语言环境)	(n.)确定整理顺序、字符类型、货币格式以及日期和时间格式的一种设置, 用于表示特定地域、文化或习俗的用户的数据。语言环境包括如何解释、存储或整理特定语言的数据的信息。语言环境还指示应使用哪个代码页表示一种特定语言。
local interface (local 接口)	(n.)一种接口, 该接口为位于同一台具有会话或实体 bean 的 Java™ 虚拟机 (Java Virtual Machine, JVM™ 机器) 中的客户机提供了访问该 bean 的机制。
Local Mail Transfer Protocol (本地邮件传输协议)	参见 LMTP 。
local part (本地部分)	(n.)电子邮件地址中标识收件人的部分。另参见 domain part (域部分) 。
local session (本地会话)	(n.)仅对一台服务器可见的用户会话。
local subset (本地子集)	(n.) DTD 的组成部分, 定义在当前 XML 文件中。
local transaction (本地事务)	(n.)源于一个数据库并且仅限于单个进程内的事务。本地事务仅对单一后端有效。本地事务通常使用 JDBC™ API 进行界定。另参见 global transaction (全局事务) 。
log directory (日志目录)	(n.)保存一个服务的所有日志文件的目录。
log expiration (日志失效期)	(n.)当日志文件达到允许的最长使用期限后, 将从日志目录中删除。
logical architecture (逻辑体系结构)	(n.)一种设计, 描绘了分布式应用程序的逻辑构件以及这些构件之间的关系 (或接口)。逻辑体系结构包括分布式 application component (应用程序组件) 以及支持这些组件所需的基础结构服务组件。
logical host (逻辑主机)	(n.)Messaging Server 2.0 (最低版本) 中的一个概念, 其中包括应用程序、应用程序数据所在的磁盘集或磁盘组以及用于访问群集的网络地址。SunPlex™ 系统中不再有此概念。

log rotation (日志轮转)

(n.)创建新日志文件作为当前日志文件的过程。所有后续记录的事件都将写入新的当前文件中。以前的日志文件不再记录事件，但仍然保留在日志目录中。

lookup (查找)

(n.)与搜索相同，使用指定的参数来筛选数据。

M

- mailbox** (邮箱) (n.)存储和查看消息的位置。另参见 [folder](#) (文件夹) 。
- mail client** (邮件客户机) (n.)帮助用户发送和接收电子邮件的程序。邮件客户机是各种网络和邮件程序的组成部分，用户与其有很多联系。邮件客户机创建和提交要传送的消息，检查新的收到的邮件，并接受和组织收到的邮件。
- mail exchange record** (邮件交换记录) 参见 [MX record](#) (MX 记录) 。
- mailing list** (邮件列表) 参见 [mail list](#) (邮件列表) 。
- mailing list owner** (邮件列表所有者) 参见 [mail list owner](#) (邮件列表所有者) 。
- mail list** (邮件列表) (n.)电子邮件地址列表，可利用邮件列表地址向这些地址发送消息。有时称为组。
- mail list owner** (邮件列表所有者) (n.)拥有向邮件列表添加成员或从其中删除成员的管理权限的用户。
- mail relay** (邮件转发器) (n.)从 [user account](#) (用户帐户) 或 [MTA](#) 接受邮件，然后将其转发给邮件收件人的消息存储或其他路由器的邮件服务器。
- mail router** (邮件路由器) 参见 [mail relay](#) (邮件转发器) 。
- managed bean creation facility** (受管 bean 创建工具) (n.)一种机制，用于定义 [JavaServer Faces technology](#) ([JavaServer Faces 技术](#)) 应用程序中使用的 [JavaBeans component](#) ([JavaBeans 组件](#)) 的特性。
- managed object** (管理对象) (n.)构成 [MIB](#) 组成部分的 [SNMP](#) 数据元素。在 [Directory Server](#) 中，管理对象存放在 `cn=monitor` 中，并且 [SNMP](#) 代理向网络管理站提供这些对象。与 [LDAP](#) 属性相同，每个受管理对象都有一个点号形式的名称和对象标识符。
- managed role** (受管理角色) (n.)允许创建显式的成员枚举列表。

management information base (管理信息库)	参见 MTA 。
management rule (管理规则)	(n.)将自定义的自调整、自配置或自恢复操作与 Application Server 中的触发事件关联起来。另参见 event (事件) 。
mapping (映射)	(1) (n.)将面向对象的模型与数据关系模型 (通常是关系数据库的模式) 关联的能力。将一个模式转换成不同结构的过程。 (2) (n.)用户到安全角色的映射应用。
mapping tree (映射树)	(n.)将后缀名称 (子树) 与数据库关联的数据结构。
master agent (主代理)	参见 SNMP master agent (SNMP 主代理) 。
master channel program (主通道程序)	(n.)通常启动向远程系统进行传输的通道程序。另参见 slave channel program (从通道程序) 。
master directory server (主目录服务器)	(n.)含有要复制的数据的读写目录服务器。
matching category (匹配类别)	(n.)与作为搜索提交的结果而返回的搜索查询相匹配的类别。
matching document (匹配文档)	(n.)与作为搜索提交的结果而返回的搜索查询相匹配的文档。
matching rule (匹配规则)	(n.)关于服务器在搜索操作期间如何比较字符串的指导。在国际搜索中, 匹配规则使服务器知道使用何种整理顺序和运算符。
MD5	(n.)由 RSA Data Security 开发的消息摘要算法。MD5 可用于生成极有可能唯一的简短数据摘要。
MD5 signature (MD5 签名)	(n.)由 MD5 算法生成的消息摘要。
MDB	(message-driven bean, 消息驱动 bean) (n.)作为异步消息使用方的企业 bean。消息驱动的 bean 不含特定客户机的状态, 但其实例变量可能含有跨客户机消息处理状态, 包括开放式数据库连接和对基于 EJB™ 体系结构的对象的对象引用。客户机为将消息驱动的 bean 作为消息侦听器的目的地发送消息, 以此方式来访问该消息驱动的 bean。
member (成员)	(n.)接收以邮件列表作为地址的电子邮件副本的用户或组。另参见 mail list (邮件列表) 、 expansion (扩散) 、 moderator (仲裁员) 。
message (消息)	(1) (n.)电子邮件的基本组成部分, 它由 header (邮件头) 和 body (正文) 组成, 并且通常在从发件人向收件人传输的过程中包含在一个 envelope (信封) 中。 (2) (n.)在 Java 消息服务中, 由 JMS 客户端使用的异步请求、报告或事件。消息包含消息头 (可以向其中添加附加字段) 和正文。消息头指定标准字段和可选属性。消息正文含有要发送的数据。消息包含协调企业应用程序所需的重要信息, 其形式为精确格式化的、描述特定业务操作的数据。

message access services (消息访问服务)	(n.)支持客户机访问 Messaging Server 消息存储的协议服务器、软件驱动程序和库。
message consumer (消息使用方)	(n.)由 JMS session (JMS 会话) (用于接收发送到 destination (目的地) 的 message (消息)) 创建的对象。
message delivery (消息传送)	(n.)当 MTA 将消息传送到本地收件人 (邮件文件夹或程序) 时发生的操作。
message-driven bean (消息驱动 bean)	参见 MDB。
message forwarding (消息转发)	(n.)当 MTA 将已传送到特定帐户的消息发送到该帐户属性指定的一个或多个新目的地时发生的操作。转发可由用户配置。另参见 message delivery (消息传送) 、 message routing (消息路由选择) 。
message handling system (消息处理系统)	参见 MHS。
message producer (消息生成方)	(n.)由 JMS session (JMS 会话) (用于将 message (消息) 发送到 destination (目的地)) 创建的对象。
Message Queue	(n.)实现 Java™ 消息服务 (Java Message Service, JMS) 开放标准的消息传送系统。Sun Java System Message Queue 是 JMS 提供者。
message queue (消息队列)	(n.)一种目录, 从客户机或其他邮件服务器接收的消息在该目录中排队, 等待立即或延迟传送。
Message Queue client runtime (Message Queue 客户机运行时环境)	(n.)一种软件, 为 JMS 客户端提供与 Java Enterprise System 消息服务器的接口。客户机运行时环境支持客户机向目的地发送消息以及从这些目的地接收消息所需的全部操作。
Message Queue message server (Message Queue 消息服务器)	(n.)为 Message Queue 消息传送系统提供传送服务 (包括与 JMS 客户端的连接、消息路由选择和传送、持久性、安全性和日志记录) 的软件。消息服务器维护的物理目的地接收 JMS 客户端发送的消息, 而该目的地的消息被发送至使用该消息的客户端。
message quota (消息配额)	(n.)一种限制, 定义特定文件夹可使用的磁盘空间大小。
message routing (消息路由选择)	(n.)当第一个 MTA 确定收件人不是本地帐户但可能存在于其他位置时, 将消息从一个 MTA 传输到另一个 MTA 的操作。路由选择通常只能由网络管理员配置。另参见 message forwarding (消息转发) 。
message selector (消息选择器)	(n.)使用方根据 JMS 消息头中的属性值 (选择器) 选择消息的方法。消息服务根据消息选择器中置入的条件执行消息过滤和路由选择。
message service (消息服务)	参见 Message Queue message server (Message Queue 消息服务器) 。
message store (消息存储)	(n.)Messaging Server 实例的所有本地传送消息的数据库。消息可以存储于单个物理磁盘上, 也可以跨多个物理磁盘存储。

- message store administrator (消息存储管理员)** (n.)拥有管理某个 Messaging Server 安装的消息存储的管理权限的用户。该用户可以查看和监视邮箱，并指定对该存储的访问控制。该用户可以使用代理授权权限运行某些实用程序，以管理消息存储。
- message store partition (消息存储分区)** (n.)驻留在单个物理文件系统分区中的消息存储或消息存储的子集。
- message submission (消息提交)** (n.)客户机 [userAgent](#) 将消息传输给邮件服务器，并请求传送。
- message transfer agent (消息传输代理)** 参见 [MTA](#)。
- messaging (消息传送)** (n.)由企业应用程序使用的一种异步请求、报告或事件系统，该系统可以使松散耦合应用程序可靠而安全地传输信息。
- Messaging Multiplexor (消息传送多路复用器)** 参见 [MMP](#)。
- Messaging Server administrator (Messaging Server 管理员)** (n.)拥有 Messaging Server 实例的安装和管理等权限的管理员。
- messaging server base directory (消息传送服务器基目录)** (n.)用于安装所有与特定主机上的特定 Administration Server 关联的服务器的目录。通常指定为 *msg_svr_base*。另参见 [installation directory \(安装目录 \)](#)。
- Messenger Express** (n.)一种邮件客户机，使用户可以通过基于浏览器 (HTTP) 的界面访问他们的邮箱。消息、文件夹和其他邮箱信息以 HTML 格式在浏览器窗口中显示。另参见 [webmail \(Web 邮件 \)](#)。
- Messenger Express Multiplexor** (n.)起到“多路复用器”作用的代理消息传送服务器。该服务器允许连接到 Messaging Server (Messenger Express) 的 HTTP 服务。Messenger Express Multiplexor 可帮助跨多个服务器机器分配邮件用户。
- metadata (元数据)** (n.)关于组件的信息，如组件名称、组件行为规范等。另参见 [deployment descriptor \(部署描述符 \)](#)。
- metadevice state database replica (元设备状态数据库副本)** (n.)存储在磁盘上、记录配置和所有元设备状态以及错误条件的数据库。该信息对 Solstice DiskSuite™ 软件磁盘集的正确运行很重要。
- metainformation (元信息)** (n.)关于资源的信息，如作者名称、文档标题、创建日期等。“搜索引擎”robot 在创建资源说明时使用元信息和文档内容。
- method-binding expression (方法绑定表达式)** (n.)[JavaServer Faces expression language \(JavaServer Faces 表达式语言 \)](#) 中引用了“支持 bean”的方法的表达式。对于其标记会使用方法绑定表达式的 UI 组件，该方法为其执行事件处理、验证或导航处理。

method permission (方法权限)	(n.)一种 authorization (授权) 规则，用于确定允许谁执行一个或多个 enterprise bean (企业 bean) 方法。
MHS	(message handling system, 消息处理系统) (n.)一组已连接的 URL mapping (URL 映射) 、其用户代理及消息存储。
MIB	(management information base, 管理信息库) (n.)一种树形结构，定义 SNMP master agent (SNMP 主代理) 可以访问的变量。MIB 提供对 HTTP 服务器的网络配置、状态和统计信息的访问。使用 SNMP 可以从 NMS 查看该信息。另参见 AUTH 。
migration (迁移)	(n.)将数据文件（如数据配置或自定义）从产品的一个版本迁移到另一个版本的过程。
MIME	(multipurpose internet mail extensions, 多用途 Internet 邮件扩展) (n.)一种多媒体电子邮件和消息传送的新标准。一种可以通过将多媒体文件附加到消息中从而将多媒体加入电子邮件消息的协议。
MIME data type (MIME 数据类型)	(n.)MIME 类型控制系统支持的多媒体文件类型。
mime.types file (mime.types 文件)	(n.)MIME 类型配置文件。此文件将文件扩展名映射到 MIME 类型，从而使服务器可以确定请求内容的类型。例如，对扩展名为 .html 的资源发出请求表示客户机正在请求 HTML 文件，而对扩展名为 .gif 的资源发出请求表示客户机正在请求 GIF 格式的图形文件。
mirror node (镜像节点)	(n.)包含与另一个活动节点相同的数据但驻留在其他数据冗余单元的活动 HADB 节点。每个活动节点必须有一个镜像节点，因此节点都是成对出现的。当一个节点检测到其镜像节点发生故障时，会接管该故障节点的角色，然后继续提供服务。另参见 HADB 、 active node (活动节点) 、 spare node (备用节点) 和 data redundancy unit (数据冗余单元) (DRU) 。
mixed-content model (混合内容模型)	(n.)一种 DTD 规范，定义某个元素包含文本和多个其他元素等混合内容。该规范必须以 #PCDATA 开头，后跟不同元素，并且必须以“零或多个”星号 (*) 结尾。
MMP	(Messaging Multiplexor, 消息传送多路复用器) (n.)一种专用 Messaging Server，它可与多个邮件服务器单点连接，从而帮助跨多个邮箱主机分配大的用户库。
mobile application configuration (移动应用程序配置)	(n.)一种 Access Manager 服务，该服务允许设置用于向移动设备传送的通讯录、日历和邮件应用程序。
mobile client type (移动客户机类型)	参见 *client type (客户机类型) 。
mobile device (移动设备)	(n.)便携式无线设备，如移动电话或个人数字助理。
mobile devices link (移动设备链接)	(n.)在“Portal 桌面”上显示的超文本链接。
mobile devices page (移动设备页)	(n.)允许用户管理移动设备选项的 Web 页面。

Mobile Portal Desktop (移动 Portal 桌面)	(n.)在移动设备上显示的“Portal 桌面”。
moderator (仲裁员)	(n.)首先接收所有目标地址包含在邮件列表中的电子邮件，以决定是否应该将该消息转发到邮件列表中的地址的人。将消息转发给邮件列表之前，仲裁人可以对该消息进行编辑。另参见 mail list (邮件列表) 、 expansion (扩散) 、 member (成员) 。
module (模块)	(1)(n.)参见 J2EE module (J2EE 模块) 。 (2)(n.)一组 Java Enterprise System 服务器，它们彼此相互依赖，或者密切相关到足以部署为一个单元，以提供特定的一种或一组服务。服务模块是经过预先测试的多服务器集合体，适于在 部署体系结构 中使用。
modutil	(n.)为外部加密设备或硬件加速器设备安装 PKCS#11 模块所需的软件实用程序。
MTA	(message transfer agent, 消息传输代理) (n.)用于路由选择和传送消息的专用程序。各 MTA 之间协同工作以传输消息并将消息最终传送至目标收件人。MTA 确定是将消息传送到本地消息存储器，还是发送到另一个 MTA 进行远程传送。
MTA configuration file (MTA 配置文件)	(n.)一种 <code>imta.cnf</code> 文件，其中包含 Messaging Server 的所有 channel (通道或频道) 定义以及确定如何为路由选择重写地址的 rewrite rule (重写规则) 。
MTA directory cache (MTA 目录高速缓存)	(n.)MTA 处理消息时需要的有关用户和组的目录服务信息快照。另参见 directory synchronization (目录同步) 。
MTA hop (MTA 转发)	(n.)在 MTA hop (MTA 转发) 之间路由消息的操作。
MUA	参见 user agent (用户代理) 。
multihomed host (多宿主主机)	(n.)位于多个公共网络上的主机。
multihost disk (多主机磁盘)	(n.)在物理上连接到多个节点的磁盘。
multimaster replication (多主复制)	(n.)一种复制模型，可以在若干主复制副本中的任一副本中写入和更新条目，而且在执行写入或更新之前不必与其他主副本通信。每个服务器都保存副本的更改日志。在一台服务器上所做的修改会自动复制到其他服务器上。出现冲突时，使用时间戳确定哪台服务器拥有最新版本。
multiplexor (多路复用器)	(n.)包含与远程服务器通信的数据库链接的服务器。
multipurpose internet mail extensions (多用途 Internet 邮件扩展)	参见 MIME 。
mutual authentication (相互验证)	(n.)一种 authentication (验证) 机制，由双方用以证明每一方相对另一方的身份。
MX record (MX 记录)	(mail exchange record, 邮件交换记录) (n.)一种 DNS 记录，它将一个主机名映射到另一个主机名。

N

n + 1 directory problem (n + 1 目录问题)	(n.)在不同类型的目录和数据库中管理同一信息的多个实例时存在的问题，导致硬件和人力成本增加。
name collision (名称冲突)	(n.)复制期间添加或重命名多个条目并尝试使用相同的 DN 时出现的冲突。目录服务器自动重命名冲突的条目以确保 DN 的唯一性。
name identifier (名称标识符)	(n.)一种假名，用于在大量服务和身份认证提供者组织之间映射用户的帐户信息，其目的是保持匿名。通过使用此标识符，身份认证提供者和服务提供者都不知道用户的真实身份。
name resolution (名称解析)	(n.)将 IP address (IP 地址) 映射到对应名称的过程。另参见 DNS 。
namespace (名称空间)	(1) (n.)LDAP 目录的树结构。另参见 DIT 。 (2) (n.)一种标准，让您能为 DTD 所定义的元素名称集指定唯一的标签。可将使用该 DTD 的文档加入到其他任何文档中，而不会引起元素名称间的冲突。然后就可以对 DTD 中定义的元素进行唯一标识，这样一来解析程序便可以确定何时应按您的 DTD 而不是按另一 DTD 中元素 <name> 的定义来解释元素 <name>。
naming attribute (命名属性)	(n.) DIT 标识名中的最终属性。另参见 relative distinguished name (相对标识名)。
naming context (命名上下文)	(1) (n.)某一 DIT 的特定后缀，由其 DN 确定。在 Directory Server 中，特定类型的目录信息存储在命名上下文中。例如，存储 Example Corporation Boston 办事处市场部员工所有条目的命名上下文可能称为 <code>ou=mktg, ou=Boston, o=example, c=US</code> 。 (2) (n.)唯一友好名称与资源之间的一组关联。另参见 JNDI extension (JNDI 扩展)、 JNDI name (JNDI 名称)、 resource (资源)。
naming environment (命名环境)	(n.)一种机制，允许在无需访问或更改组件源代码的情况下对组件进行自定义。容器将实现组件的命名环境并将其作为 JNDI naming context (命名上下文) 提供给组件。每个组件都使用 <code>java:comp/env</code> JNDI 上下文来命名和访问其环境条目。环境条目在组件的部署描述符中明确指定。

- native channel (本机频道)** (n.)显示本机内容的 Portal Server 频道。
- native content (本机内容)** (n.)使用本机标记语言 (例如 HTML) 编写的内容, 无需转换即可发送到客户机。
- native desktop (本机桌面)** (n.)显示本机内容的 Portal Server Desktop。
- NDN** (nondelivery notification, 邮件未发送通知) (n.)消息传送期间 MTA 在 **address (地址)** 和 **rewrite rule (重写规则)** 之间找不到匹配项时由 MTA 发回给发件人的无法传送报告 (附原始消息)。
- nested role (嵌套角色)** (n.)命名其他角色定义的 **role (角色)**。某嵌套角色的成员集是其所包含角色的所有成员的并集。嵌套角色还可以定义扩展范围, 以便将其他子树中角色的成员包括在内。
- NetFile** (n.)一种基于 Java™ 技术的文件服务器应用程序, 使用户可以远程访问文件系统, 从而允许对文件和目录进行远程操作。
- Netlet** (n.)Java Enterprise System Portal Server 中使用的一种 Java applet, 允许任何基于传输控制协议/Internet 协议的应用程序通过已验证的 Portal Server 连接安全地连接到服务器。
- NetMail** (n.)NetMail 组件实现 NetMail (基于 Java 技术的客户机) 和 NetMail Lite 电子邮件客户机。这些客户机与标准 IMAP 和 SMTP 服务器一起工作。
- Netscape™ Console (Netscape 控制台)** (n.)使用 Java 编程语言编写的一个应用程序, 它为服务器管理员提供一个图形界面, 用于在企业网络内任一中心位置管理所有 Netscape 服务器。通过安装的任何 Netscape Console 实例, 可以查看和访问企业网络上已授予您访问权限的所有 Netscape 服务器。
- network management station (网络管理站)** 参见 **NMS**。
- network manager (网络管理器)** (n.)用于读取、格式化和显示 **SNMP** 数据的程序。又称为 SNMP 客户机。
- Network Security Services for Java, JSS (面向 Java 的网络安全服务)** (n.)提供到 Network Security Services SSL 库的 Java 绑定的类库。Portal Server 使用该类库启动来自 servlet 的 **secure socket layer (安全套接字层)** 连接和接受 Portal Server Secure Remote Access Pack 网关中的 SSL 连接。
- news channel (新闻频道)** (n.)用于发布和共享信息的论坛。用户订阅新闻频道的目的是查看更新。新闻频道中的信息通常使用 URL 自动发布, 或者由具有适当权限的用户发布。管理员可以控制新闻频道访问, 方式是将用户分配到他们需要的频道并且决定谁可以查看新闻频道中的信息或向新闻频道发布信息。
- news channel list (新闻频道列表)** (n.)显示您当前订阅的所有新闻频道的一个窗口。每个新闻频道都用单独的选项卡表示。
- next-hop list (下一中继站列表)** (n.)相邻系统列表, 邮件路由使用它确定向何处传输消息。下一中继站列表中系统的顺序决定邮件路由向这些系统传输消息的顺序。

NIS	(network information service , 网络信息服务) (n.) (仅限 UNIX) 一种关于程序和数据文件的系统, UNIX 系统使用它在某一计算机网络中收集、整理和共享有关计算机、用户、文件系统和网络参数的特定信息。
NMS	(network management station , 网络管理站) (n.) 安装了一个或多个网络管理应用程序的功能强大的工作站。NMS 是用于远程管理您的网络的计算机。
NNTP	(Network News Transfer Protocol , 网络新闻传输协议) (n.) 一种新闻组协议。要在服务器上使用代理服务, 必须定义新闻服务器主机。
node (节点)	(1) (n.) 计算节点。网络或 Internet 环境中的诸多计算机之一。分布式应用程序部署在此环境中, 其中不同的分布式组件、 业务服务 和 服务器 运行于各种计算节点上。另参见 cluster (群集) 。 (2) (n.) 参见 HADB node (HADB 节点) 。
node agent (节点代理)	(n.) 一种轻量代理, 每台至少托管一个 Application Server server instance (服务器实例) 的机器都需要此代理, 包括托管 Domain Administration Server (域管理服务器) 的机器。节点代理按照 Domain Administration Server 的指示执行任务, 包括启动、停止、创建和删除 Application Server 实例。
nondelivery notification (邮件未发送通知)	参见 NDN 。
NoPassword authentication (无密码验证)	(n.) 一种验证类型, 允许用户登录 Access Manager 而不提示其输入密码。
normalization (标准化)	(n.) 通过模块化来删除冗余 (如同子例行程序) 以及通过将多余差异缩减为一个通用标准来消除差异的过程。例如, 对来自不同系统的行尾, 通过将它们缩减为单个新的行来对其进行标准化, 而多个空白字符经过标准化后成为一个空格。
North American Industry Classification System, NAICS (北美产业分类系统)	(n.) 根据营业机构所生产产品或提供服务的过程来对其进行分类的系统。
NOTARY messages (NOTARY 消息)	(n.) 符合 NOTARY 规范 RFC 1892 的邮件未发送通知 (Nondelivery notification, NDN) 和传送状态通知。
notation (表示法)	(n.) 一种定义数据格式的机制, 作为未解析实体来引用的非 XML 文档的数据格式是以此机制进行定义。这是来自 SGML 的延续。一种更新的标准将采用 MIME 数据类型和名称空间来避免命名冲突。
notification message (通知消息)	(n.) 一种由 Messaging Server 发送的消息, 它提供消息传送处理的状态及任何传送问题或完全失败的原因。这些消息只是为了提供信息, 不需要邮寄主管执行操作。另参见 delivery status notification (传送状态通知) 。
notification service (通知服务)	(n.) 一种从其他服务器接收订阅和通知, 然后将通知转发给特定订户的服务。Calendar Server csnotifd 服务使用“事件通知服务” (Event Notification Service, ENS) 作为事件代理来发送事件和待办任务的通知。

NSAPI	参见 server plug-in API （服务器插件 API）。
ns-slapd	(n.)（仅限 UNIX）负责 Directory Server 所有操作的一个进程或服务。Windows 系统中与之相当的是 slapd.exe 。
ns-slapd.exe	(n.)（仅限 Windows）Windows 系统中的进程监视器。

O

- OASIS** (Organization for the Advancement of Structured Information Standards, 结构化信息标准促进组织) (n.)推动电子商务标准发展、融合和应用的联盟。其 Web 站点为 <http://www.oasis-open.org/>。它所发起的 DTD 系统信息库位于 <http://www.XML.org>。
- obj.conf file (obj.conf 文件)** (n.)服务器的对象配置文件。此文件包含附加的初始化信息、服务器自定义设置, 以及服务器处理来自客户机(例如浏览器)的请求时使用的指令。Web Server 每次处理客户机请求时都读取此文件。
- object class (对象类)** (n.)一种模板, 用于指定条目所描述的对象种类和条目所包含的属性集。例如, Directory Server 指定一个 emailPerson 对象类, 它具有诸如 commonname、mail (电子邮件地址)、mailHost 和 mailQuota 这些属性。
- object identifier (对象标识符)** 参见 **OID**。
- object persistence (对象持久性)** 参见 **persistence (持久性)**。
- obsolete patch (过时修补程序)** (n.)不再视为有效或最新的修补程序。某修补程序被视为过时的三种情况: 该修补程序的后续版本修复相同的问题、另一个修补程序包含原修补程序的修复补丁, 或者该修补程序不再视为相关。
- offline state (脱机状态)** (n.)一种状态, 在此状态下, 邮件客户机从服务器系统将消息下载到客户机系统, 可以在客户机系统中查看和回复消息。这些消息可能从服务器删除, 也可能不删除。
- OID** (object identifier, 对象标识符) (n.)对象标识符的字符串表示, 由使用句点分隔的十进制数组成(例如, 1.3.6.1.4.1)。在 **Lightweight Directory Access Protocol (轻量目录访问协议)** 中, 对象标识符用于唯一标识模式元素, 包括对象类和属性类型。对象标识符分层结构的顶层由标准主体管理, 并委托给希望构造自己的模式定义的组织。
- OMG** (Object Management Group, 对象管理组织) (n.)一个联盟, 它制定并维护关于可交互使用的企业应用程序的计算机行业规范。其 Web 站点为 <http://www.omg.org/>。
- one-way messaging (单向消息传送)** (n.)一种发送 **message (消息)** 的方法, 该方法不必在接收到响应前一直阻塞。

online state (联机状态)	(n.)一种状态, 在此状态下, 消息保留在服务器上并由邮件客户机远程响应。
operational attribute (操作属性)	(n.)操作属性包含目录在跟踪修改以及子树属性时内部使用的信息。操作属性不会在对搜索的响应中返回, 除非显式请求。
optional attribute list (可选属性列表)	(n.)某一指定对象类的可选属性列表。可选属性有前导关键字 MAY。
ORB	(Object request broker, 对象请求代理) (n.)使 CORBA 对象能找到彼此并相互进行通信的库。
organization (组织)	(n.)Directory Server Access Management Edition 中的一个对象, 代表层次结构的顶级, 企业使用它来管理部门和资源。安装 Directory Server Access Management Edition 时, 它会自动创建一个顶层组织 (default o=isp) 来管理 Directory Server Access Management Edition 企业配置。安装之后, 可以创建其他组织来管理单独的企业。创建的所有组织都位列顶层组织之下。另参见 suborganization (子组织) 。
organization administrator (组织管理员)	(n.)拥有以下管理权限的用户: 可通过用于消息传送和协作的 Delegated Administrator GUI 或 CLI, 在组织或子组织中创建、修改和删除邮件用户和邮件列表。
O/R mapping tool (O/R 映射工具)	(object-to-relational database tool, 对象关系数据库工具) (n.)Application Server 管理界面内的一个映射工具, 可为实体 bean 创建 XML 部署描述符。
OSI tree (OSI 树)	(Open Systems Interconnect tree, “开放式系统互连”树) (n.)一个反映“开放式系统互连”网络语法的 DIT 。OSI 树中的一个标识名 (distinguished name, DN) 示例为 cn=billt,o=bridge,c=us。
OS principal (OS 主体)	(n.)运行 J2EE 平台的操作系统的本机主体。
OTS	(Object Transaction Service, 对象事务服务) (n.)对允许 CORBA 对象参与事务的接口的定义。

P

- package (软件包)** (n.)文件与目录的集合。封装软件包是分发安装软件的一种方法。另参见 [assembly \(装配\)](#)、[deployment \(部署\)](#)。
- parameter (参数)** (1) (n.)自 Java Enterprise System Application Server 客户机发送的名称值对，包括表单字段数据、HTTP 头信息等，它封装在一个请求对象中。另参见 [attribute \(属性\)](#)、[property \(属性\)](#)。
(2) (n.)Java 方法或数据库预处理命令的参数。
- parameter entity (参数实体)** (n.)由 [DTD](#) 规范组成的实体，与一般实体不同。在 DTD 中定义参数实体可在其他点引用，从而无需在每个使用位置都对其定义进行重新编码。
- parent (父)** (n.)XML 文件中的一个元素，它包含被作为子来引用的另一个元素。另参见 [child \(子\)](#)。
- parent access (父访问权限)** (n.)如果授予，则表示当 [bind DN \(绑定 DN\)](#) 是目标条目的父项时，用户具有对目录树中自身位置之下的条目的访问权限。
- parsed entity (已解析实体)** (n.)一般性实体，它包含 [XML](#)，并因此在插入到 XML 文档中后得以解析，区别于未解析实体。
- parser (解析程序)** (n.)一种模块，它从输入源读入 [XML](#) 数据并将其分解成块，这样程序就知道它何时使用标记、属性或元素数据。非验证解析程序能确保 XML 数据的格式正确，但不会检验它的有效性。另参见 [validating parser \(验证解析程序\)](#)。
- partition (分区)** 参见 [message store partition \(消息存储分区\)](#)。
- passivation (钝化)** (n.)将 [enterprise bean \(企业 bean\)](#) 从内存转移到辅助存储器的过程。从内存释放某一 bean 的资源而不损坏该 bean 的一种方法。这样，可令 bean 持久，无需实例化系统开销即可重新调用。另参见 [activation \(活化\)](#)。
- pass-through authentication (传递验证)** 参见 [PTA](#)。
- pass-through subtree (传递子树)** (n.)在传递验证中，PTA Directory Server 将来自其 [DN](#) 包含在此子树中的所有客户机的绑定请求传递到验证 Directory Server。

password authentication (密码验证)	(n.)通过用户名和密码确认用户身份。另参见 certificate-based authentication (基于证书的验证)。
password file (密码文件)	(n.) (仅限 UNIX) 存储 UNIX 用户登录名、密码和用户 ID 号的文件。该密码文件因其所在位置又称为 <code>/etc/passwd</code> 。
password policy (密码策略)	(n.)控制如何在给定目录中使用密码的一组规则。
patch version number (修补程序版本号)	(n.)修补程序标识符的后两位数字, 例如 "nnnnnn-03"。每次发布修补程序新版本时, 此编号都加一。
pattern (模式)	(n.)用于匹配目的的字符串表达式, 如“允许”和“拒绝”过滤器中的字符串表达式。
PCDATA	(n.)一种用于已解析字符数据的预定义 XML 标记, 适用 XML 语法的标准规则, 它区别于字符数据 (CDATA), 后者意味着“不解释这些字符。”另参见 CDATA 。
PDC	(personal digital certificate, 个人数字证书) (n.)附加到消息的、用于验证用户身份的电子证书。可以通过正确输入用户 ID 和密码创建个人数字证书, 也可以使用 SSL certificate (SSL 证书) 请求 (该请求又将使用连接用户时所经服务器的安全性证书) 来创建个人数字证书。
peer (同级类别)	(n.)具有与另一类别相同父类别的子类别。
permanent failure (永久性故障)	(n.)消息处理过程中出现的错误情况。出现永久性故障时, 消息存储器会删除其电子邮件消息副本。 MTA 将消息发回给发件人并删除其消息副本。
permissions (权限)	(1) (n.)授予或拒绝给某一用户或组的一组权限。此信息包括用户名或组名、一个或多个有效电子邮件地址, 以及如何和向何处传送电子邮件。 (2) (n.)在访问控制上下文中, 权限声明是授予还是拒绝访问目录信息的权限, 以及授予或拒绝的访问权限级别。另参见 access rights (访问权限)。 (3) (n.)控制对日历的访问权限的设置。例如, 在 Calendar Express 中, 权限包括 Availability (可用性)、 Invite (邀请)、 Read (读取)、 Delete (删除) 和 Modify (修改)。 Calendar Server 管理员使用命令行实用程序将权限设置为 ACE 字符串。另参见 ACL 。
persistence (持久性)	(1) (n.)对于组件, 指实例变量和底层数据库之间的状态转移协议。参见 entity bean (实体 bean)。另参见 transience (瞬态)。 (2) (n.)对于会话而言, 指会话存储机制。另参见 session (会话)、 failover (故障转移)、 session failover (会话故障转移)。
persistence manager (持久性管理器)	(n.)负责 EJB 1.x 或 2.x entity bean (实体 bean) 持久性的管理器。
persistent field (持久性字段)	(n.)EJB 2.1 entity bean (实体 bean) 的虚拟字段, 具有 container-managed persistence (容器管理持久性); 它存储在数据库中。
persistent state (持久状态)	(n.)其中, 对象状态保存在持久存储器中 (通常为数据库)。

personal digital certificate (个人数字证书)	参见 PDC 。
personal folder (个人文件夹)	(n.)只能由所有者读取的文件夹。另参见 shared folder (共享文件夹)
pk12util	(n.)从您的内部计算机导出证书和密钥数据库并将它们导入外部 PKCS#11 模块所需的软件实用程序。
PKI	(public key infrastructure , 公钥基础结构) (n.)使用户的身份能链接到浏览器或移动设备。“无线 PKI”指出现在手提设备上的 certificate-based authentication (基于证书的验证)。
plaintext (纯文本)	(n.)传送数据的一种方法。其定义依赖于上下文。对于 secure socket layer (安全套接字层), 纯文本密码会被加密, 因此不会作为明文发送。对于 SASL , 纯文本密码将被散列, 并且仅将密码的一个散列作为文本发送。
plaintext authentication (纯文本验证)	参见 password authentication (密码验证)。
pluggable authentication (可插接式验证)	(n.)允许 J2EE 应用程序从 J2SE™ 平台使用 Java™ 验证和授权服务 (Java Authentication and Authorization Service, JAAS) 软件的一种机制。开发者可以插入他们自己的验证机制。
plug-in (插件)	(1) (n.)一种对浏览器的代码扩展, 可显示或执行 Web 页面中的内容。插件使浏览器能够显示浏览器在未使用插件时不能显示的页面内容元素。 (2) (n.)一种可以加载, 然后作为整体系统的一部分使用的辅助程序。例如, Calendar Server 可以使用插件访问非 LDAP 目录服务。
POA	(Portable Object Adapter , 可移植对象适配器) (n.)一种 CORBA 标准, 适用于构建可以在各种 ORB 上移植的服务器端应用程序。
pointer CoS (指针 CoS)	(n.)一种只使用模板 DN 标识模板条目的指针类服务。
point-to-point delivery model (点对点传送模型)	(n.)一种模型, 其中 message producer (消息生成方) 将 message (消息) 发送到特定 message queue (消息队列), 然后 message consumer (消息使用方) 从保存消息的队列中提取消息。一条消息只传送给一个消息使用方。
policy (策略)	(1) (n.)描述授权何人在特定条件下访问特定资源的规则。可以基于组织中的用户组或角色来建立规则。 (2) (n.)在 Directory Server Access Management Edition 中, 定义规则以帮助保护组织的 Web 资源。策略只分配给组织和角色。
poll (轮询)	(n.) Instant Messaging Server 中的一种功能, 使您能够询问用户对某一问题的回答。您可以向选定用户发送问题和可能的答案, 他们回馈所选答案。
pooling (入池)	(n.)提供许多预配置资源以提高性能的过程。如果某一资源入池, 组件可使用池中的现有实例, 而不必实例化一个新实例。在 Java Enterprise System Application Server 中, 数据库连接、 servlet 实例和企业 bean 实例都可以入池。

POP3	(Post Office Protocol Version 3, 邮局协议版本 3) (n.)一种协议, 可提供标准传送方法而且不需要 MTA 具备访问用户邮件文件夹的权限。在邮件客户机和消息传输代理通常位于不同计算机上的联网环境中, 不需要具有访问权限是一个有利条件。
port (端口)	(n.)进行传输控制协议/Internet 协议连接的位置(套接字)。Web 服务器传统上使用端口 80; FTP 使用端口 21; telnet 使用端口 23。Java Enterprise System Portal Server 使用特殊端口(尤其是在客户机系统上)通过 Portal Server 会话与服务器安全地进行通信。
portal (门户)	(n.)一组资源的入口点, 企业想要使这些资源对门户用户可用。对于某些使用者门户, 资源组包括整个万维网, 但是对于大多数企业, 资源组包括特定于用户和企业之间关系的信息、应用程序和其他资源。Portal Server Desktop 是用于在 Portal Server 中生成门户的应用程序。
Portal Desktop (Portal 桌面)	(n.)由 Portal Server 生成的任一桌面。
Portal Server	(n.)使远程用户能够通过 Internet 安全地访问其组织的网络和网络服务的软件产品。它创建一个安全的 Internet 门户, 使所有目标受众(包括员工、业务伙伴或一般公众)都能访问相应内容、应用程序和数据。被认为是整个 Sun Java System Portal Server 产品解决方案的核心部分, 由所有 Portal Server 包共享。
Portal Server Desktop	(n.)它提供主终端用户界面, 以及通过内容提供者接口(PAPI)实现可扩展内容聚合的机制。通常又称为“桌面”。“桌面”包括多种提供者, 它们提供构建某些类型频道所需的容器分层结构和基本构件。“桌面”在 Directory Server Access Management Edition 服务之上实现一种显示配置数据存储机制, 用于存储内容提供者和频道数据。“桌面”还包括一个管理控制台模块, 用于编辑显示配置及其他“桌面”服务数据。
Portal Server Instant Collaboration Pack	(n.)一种服务器即时消息传送产品, 包括服务器、multiplexor (多路复用器)和 Instant Messaging 组件。又称为 Instant Messaging Server。
Portal Server Pack	(n.)指代 Portal Server 附加产品的一个通用术语。
portal node (门户节点)	(n.)运行 Portal Server 软件或 Portal Server Pack 软件的物理计算机。又称为 host (主机)。
port number (端口号)	(n.)用于指定主机上个别传输控制协议/Internet 协议应用程序的编号。它为传送的数据提供目的地。
post-deployment (后期部署)	(n.)Java Enterprise System 解决方案生命周期过程中的一个阶段, 在此阶段, 将启动分布式应用程序, 对其进行监视、调节以优化性能, 还会对其进行动态升级以加入新的功能。
postinstallation configuration (安装后配置)	(n.)在运行 Java Enterprise System 安装程序(通常选择“以后再配置”选项)后所执行的 Access Manager 配置任务。通常, 安装后任务只需执行几次。例如, 可能需部署产品的附加实例或配置产品以用于会话故障转移。另参见 configuration (配置)。
postmaster account (邮寄主管帐户)	(n.)从 Messaging Server 接收系统生成的消息的电子邮件组和电子邮件地址的别名。邮寄主管帐户必须指向一个或多个有效邮箱。
Post Office Protocol Version 3 (邮局协议版本 3)	参见 POP3。

pre-deployment (前期部署)	(n.)Java Enterprise System 解决方案生命周期过程中的一个阶段, 在此阶段中, 业务需要被转换成 deployment scenario (部署方案) : 一种 logical architecture (逻辑体系结构) 。
preferred directory server (首选目录服务器)	(n.)目录服务器主实例, 由适用于 Windows 的身份认证服务用来检测更改并将更改应用于用户条目。该服务器可用时, 适用于 Windows 的身份认证服务不会与任何其他主目录服务器进行通信。
prepared command (预处理命令)	(n.)SQL 中的一条数据库命令, 经过预编译后会使得重复执行更加有效。预处理命令可以包含参数。另参见 prepared statement (预处理语句) 。
prepared statement (预处理语句)	(n.)封装重复用于取得数据的 QUERY、UPDATE 或 INSERT 语句的一个类。预处理语句至少包含一条 prepared command (预处理命令) 。
presence index (存在索引)	(n.)一种过滤方法, 它允许高效搜索包含特定类型的某一属性的条目, 而不必考虑条目中该属性的值如何。
presentation layout (表现布局)	(n.)Web 页面内容的格式。
presentation logic (显示逻辑)	(n.)在应用程序中创建页面的活动, 包括处理请求、生成响应内容以及格式化客户机页面。通常由 Web 应用程序处理。
preset message (预设消息)	(n.)可以编写并保存为 Portal Server Mobile Access 移动首选项, 以备将来在移动电子邮件应用程序中使用的短消息。
primary data view (主数据视图)	(n.)构成联合数据视图的两个 Directory Proxy Server 数据视图中的一个。默认情况下, 主数据视图是条目的权威性来源。另参见 secondary data view (从数据视图) 。
primary document directory (主文档目录)	参见 document root (文档根目录) 。
primary key (主键)	(n.)使客户机在主页中找到特定 EJB 2.1 entity bean (实体 bean) 的唯一标识符。
primary key class name (主键类名)	(n.)一个变量, 它指定了 bean 的主键的全限定类名。用于 Java Naming and Directory Interface™ (JNDI) 查找。
principal (主体)	(n.)作为 authentication (验证) 结果指定给用户的身份。主体可以获得联合身份, 具有这种身份可以制定决策并能自行执行验证操作。例如, 主体可以是个人用户、一组个人、公司、其他法律实体, 或者是 Liberty 体系结构的组成部分。
private key (私钥)	参见 public-key cryptography (公钥密码学) 。
privilege (权限)	(n.)授予单个用户、一组用户或单个资源的一类访问权。该安全属性并不唯一, 可以由许多主体共享。
process (进程)	(1) (n.)一种由操作系统设置的自包含、功能完全的执行环境。应用程序的每个实例通常在一个单独的进程中运行。 (2) (n.)活动程序的执行序列。一个进程由一个或多个线程组成。
processing instruction (处理指令)	(n.)包含在 XML 结构中, 旨在由特定应用程序解释的信息。

-
- produce (生产)** (v.)将消息传递给客户机运行时环境以送达目的地。
- producer (生成方)** (n.)由会话创建的对象 (MessageProducer), 用于将消息发送到目的地。在点对点传送模型中, 生成方是发送者 (QueueSender)。在发布/订阅传送模型中, 生成方是发布者 (TopicPublisher)。
- production environment (生产环境)** (n.)应用程序生命周期过程中的一个阶段, 在此阶段, 将启动分布式应用程序, 对其进行监视、性能调优, 并进行动态升级以加入新功能。
- programmatic security (程序安全性)** (n.)在代码中显式控制安全性的处理方法, 此方法不允许诸如组件的容器、bean 的容器或 servlet 引擎等进行安全处理。与 **declarative security (声明性安全)** 相对。当仅依靠声明性安全不足以表述应用程序的安全模型时, 程序安全性很有用。
- programmer-demarcated transaction (程序员区分的事务)** 参见 **bean-managed transaction (bean 管理事务)**。
- prolog** (n.)XML 文档中位于 XML 数据前的部分。prolog 包括声明以及可选的 DTD。
- propagation behavior (传播行为)** (n.)使用方与供应者间的同步过程。
- property (属性)** (1) (n.)定义应用程序组件行为的单个名称值对。另参见 **parameter (参数)**。
(2) (n.)用于修改 XML 文件中的元素但又**没有**在 DTD 文件中预定义的名称值对。与 **attribute (属性)** 相对。
(3) (n.)在 Application Server 中, **不是**内置服务器配置一部分的名称值对。与 **attribute (属性)** 相对。
- protocol (协议)** (1) (n.)描述网络设备如何交换信息的一组规则。
(2) (n.)对要交换的消息以及两个或多个系统交换信息时需遵循的规则进行的形式描述。
- provider (提供者)** (n.)频道的程序学形态。向提供者添加配置数据即会使其摇身变为频道实例。提供者是一个 Java 类, 负责将文件内容或是应用程序或服务的输出转换成适当的频道格式。Portal Server 随带有多个提供者, 其中包括书签提供者、应用程序提供者和备忘录提供者。桌面成像时, 会轮流查询每个提供者所关联频道的内容。一些提供者能够基于各自的配置生成多个频道。

内容提供者的示例有 UserInfoProvider 和 BookmarkProvider。**container (容器)** 提供者的示例有 TabContainerProvider 和 SingleContainerProvider。叶提供者的示例有 JSPProvider、XMLProvider、URLScrapperProvider 和 SimpleWebServicesProvider。
- provider federation (提供者联合)** (n.)一组服务提供者, 他们通过合同的形式就使用基于自由联盟计划规范的体系结构来交换验证信息达成一致。另参见 **authentication domain (验证域)**。
- provisioning (置备)** (n.)在 Java Enterprise System Directory Server 中添加、修改或删除条目的过程。这些条目包括用户、组及域信息。
- proxy (代理)** (1) (n.)一种机制, 利用该机制一个系统可代表另一系统对协议请求作出响应。在网络管理中使用代理系统, 便无需在简单设备 (如调制解调器) 中实现全部的协议栈。

	(2) (n.)代表客户机发出请求或服务请求的中介程序。代理轮流充当服务器和客户机，并用来控制各种网络服务的内容。另参见 reverse proxy (反向代理) 。
proxy authorization (代理授权)	(n.)一种特殊形式的验证，其中，客户机以自身身份绑定到目录，但基于每一操作授予其另一用户的访问权。该另一用户称为代理用户，其 DN 为代理 DN。
proxy DN (代理 DN)	(n.)一个条目的 DN，该条目拥有对客户机应用程序正要在其上执行操作的目标的访问权限。与 proxy authorization (代理授权) 一起使用。
Proxylet	(n.)一种动态代理服务器，在客户机上运行以将 URL 重定向到 SRA 网关。另参见 Secure Remote Access (SRA)
PTA	(pass-through authentication, 传递验证) (n.)一种机制，Java Enterprise System Directory Server 利用这种机制向另一 Directory Server 查询绑定规则。
PTA Directory Server	(n.)在 pass-through authentication (传递验证) 中，PTA Directory Server 将其收到的绑定请求发送（传递）给验证 Directory Server。
PTA LDAP URL	(n.)在 pass-through authentication (传递验证) 中，定义验证 Directory Server、传递一个或多个子树以及可选参数的 URL。
public folder (公共文件夹)	(n.)具有多个所有者的文件夹，由多个可对其访问的人员共享。根据为该文件夹设置的 ACL ，该文件夹可由多个人更新或管理。
public information directories (公共信息目录)	(n.)（仅限 UNIX）文档根目录以外、位于 UNIX 用户主目录或处于用户控制下的目录。
public key (公钥)	(n.)公钥加密中使用的加密密钥。
public-key certificate (公钥证书)	(n.)一种数据结构，其中包含用户的公钥以及有关证书有效时间和日期的信息。用于客户机证书验证，以使服务器或有选择地使客户机能互相验证。公钥证书是等价的数字通行证。它由受信组织（称为证书授权机构）颁发，并向载送方提供标识。
public-key cryptography (公钥密码学)	一种加密方法。在公钥密码系统中，每个人都有两个相补密钥：一个公示的密钥和一个秘密的密钥（又称为私钥）。每个密钥可对另一密钥生成的代码进行解锁。知道公钥对于推断相应的私钥并无帮助。公钥可以公布并在通信网络上广泛散播。此协议无需传统密码系统所要求的安全通道即可提供保密功能。又称为非对称密钥密码学。
public-key encryption (公钥加密)	(n.)一种加密方法，使用由公共和私有组件组成的两部分密钥（代码）。加密邮件使用的是收件人已经公布的公钥。收件人使用只有自己知道的未公布私钥来解密邮件。
public key infrastructure (公钥基础结构)	参见 PKI 。
Public Network Management (公共网络管理)	(n.)通过故障监视和故障转移防止因单个网络适配器或电缆故障而丧失节点可用性的软件。“公共网络管理”故障转移使用称为“网络适配器故障转移组”的多组网络适配器，在群集节点与公共网络之间提供冗余连接。故障监视和故障转移功能相互配合，共同确保资源的可用性。

publish and subscribe delivery model (发布和订阅传送模型) (n.)一种消息传送系统，其中的发布者和订阅者通常为匿名，并可动态发布或订阅内容层次结构中的特定节点，这种节点称为 **topic (主题)**。该系统将从某个主题的多个发布者那里收到的 **message (消息)** 分发给它的多个订户。

purge a message (清除消息) (v.)永久除去已删除并且在用户和组文件夹中不再引用的消息。随后将空间返还给消息存储文件系统。另参见 **delete a message (删除消息)** 和 **expunge a message (清除邮件)**。

Q

- QOS** (quality of service, 服务质量) (n.)为服务器实例或虚拟服务器设置的性能限制。例如，如果您是一个 ISP，您可能希望根据所提供的带宽来收取不同的虚拟服务器费用。您可以限制带宽总量和连接数量。
- query string (查询字符串)** (n.)HTTP 请求 URL 的一个组件，包含一组会影响请求处理的参数和值。
- queue (队列)** (n.)在 Java 消息服务中，队列是管理员为实现点对点传送模型而创建的对象。队列始终可用于存放消息，即使在使用其消息的客户机处于非活动状态时亦是如此。队列被用来作为生产者与消费者之间的一个中间存放地。参见 [JMS](#)、[point-to-point delivery model \(点对点传送模型\)](#)。

R

RAF	(robot application function, robot 应用函数) (n.)一种可在 robot 过滤器配置文件中使用的函数。用户定义的 robot 应用函数又称为插件函数。这些函数通过指令调用。
RAM	(random access memory, 随机存取存储器) (n.)计算机中用半导体制成的物理存储器。
RAR file (RAR 文件)	(resource adapter archive, 资源适配器归档文件) (n.)包含资源适配器模块 (也称为 connector (连接器) 模块) 的 Java™ 归档 (Java archive, JAR) 文件。
RC2	(n.)由 RSA Data Security 开发的可变密钥大小块加密器。
rc.2.d (rc.2.d 文件)	(n.) (仅限 UNIX) UNIX 机器上的一个文件, 描述机器启动时运行的程序。此文件因其所在位置又称为 <code>/etc/rc.2.d</code> 。
RC4	(n.)由 RSA Data Security 开发的流加密器。比 RC2 速度快。
RD	参见 resource description (资源描述) 。
RDB	(n.)Relational database (关系数据库)。
RDBMS	(n.)Relational database management system (关系数据库管理系统)。
RDF	(Resource Description Framework, 资源描述框架) (n.)定义 XML 文件所包含的数据种类的标准。此类信息有助于确保语义完整性, 例如, 有助于确保将日期视为日期, 而不仅仅是文本。
RDF schema (RDF 模式)	(n.)一种标准, 用于指定应用到 RDF 中所包含的规范的一致性规则。
RDM	参见 resource description message (资源描述消息) 。
RDN	(relative distinguished name, 相对标识名) (n.)将实际条目的祖先附加到字符串末尾形成完整 DN 前, 该条目本身的名称。多数 RDN 由来自条目的单个属性类型和值组成。
read-only bean (只读 bean)	(n.)永远不会被 EJB™ 客户机修改的实体 bean。另参见 entity bean (实体 bean) 。

- realm (领域)** (n.)安全服务的安全管理员定义和实施公共安全策略的范围。也称为安全策略域或安全域。在 J2EE 服务器验证服务中, 领域是一个完整的角色、用户 (或主体) 和组的数据库, 它们将标识一个 Web 应用程序或一组 Web 应用程序的有效用户。
- redirection (重定向)** (n.)一种机制, 用以将访问特定 URL 的客户机发送到相同或不同服务器上的另一不同位置。重定向在资源移动后想要客户机透明地使用新位置时很有用。重定向还用于在不带尾随斜杠访问目录时维护相对链接的完整性。
- reentrant entity bean (可重入实体 bean)** (n.)一种 **entity bean (实体 bean)**, 可处理多个不会相互干扰的并发调用、交叉调用或嵌套调用。
- reference (引用)** (n.)解析 XML 文档时会代替引用的实体引用。参见 **entity reference (实体引用)**。
- reference deployment architecture (参考部署体系结构)** (n.)一种已设计、实现并经过性能测试的 **deployment architecture (部署体系结构)**。参考部署体系结构用来作为设计定制解决方案部署体系结构的起点。
- referential integrity (引用完整性)** (n.)一种机制, 用于确保在目录内保持 DN 取值属性所表示的条目之间的关系。
- referral (引用)** (n.)当服务器从客户机那里收到无法处理的搜索或更新请求时, 服务器会向客户机回送一个指针, 该指针指向可以处理该请求的 Java Enterprise System Directory Server。
- referral hop limit (引用跳数限制)** (n.)客户机在一行中应遵从的最大引用数。
- registry (注册表)** (n.)可构建、部署和搜索 **web service (Web 服务)** 的基础结构。它是中立的第三方, 可促进动态耦合和松散耦合的企业到企业 (business-to-business, B2B) 交互。
- registry provider (注册表提供者)** (n.)一种符合 XML 注册表规范的业务注册表 (例如 **ebXML** 或 **UDDI**)。
- regular expression (正则表达式)** (n.)一种文本字符串, 它使用特殊字符来表示字符的范围或分类, 目的是要进行模式匹配。
- relative distinguished name (相对标识名)** 参见 **RDN**。
- relationship field (关系字段)** (n.)具有 **container-managed persistence (容器管理持久性)** 的 **entity bean (实体 bean)** 的虚拟字段; 它标识相关实体 bean。
- relaying (中继)** (n.)将消息从一个消息传送服务器传递给另一消息传送服务器的过程。
- remote interface (remote 接口)** (n.)EJB 1.x 和 2.x 组件的两个接口之一。remote 接口定义可由客户机调用的业务方法。另参见 **home interface (home 接口)**。
- remove method (remove 方法)** (n.)一种在 **home interface (home 接口)** 中定义的方法, 客户机调用该方法来损坏 EJB 1.x 或 2.x 企业 bean。
- renderer** (n.)一种 Java 类, 可呈现一组 **JavaServer Faces UI component (JavaServer Faces UI 组件)** 的输出。

rendering (呈现)	(1) (n.)将采用抽象标记语言 (Abstract Markup Language, AML) 编写的内容转换为适合特定移动设备的设备特定标记语言的过程。 (2) (n.)生成客户机输出的过程。参见 renderer 。
rendering channel (呈现频道)	(n.)用于显示呈现内容的 Portal Server Mobile Access 频道。
rendering engine (呈现引擎)	(n.)在 Portal Server 中, 用于将 AML 转换为适合于给定移动客户机的语言。
rendering filter (呈现过滤器)	(n.)用于在呈现引擎与客户机间传递所要转换内容的过滤器。
render kit (呈现工具包)	(n.)一组可将输出呈现到特定客户机的 renderer 。 JavaServer Faces technology (JavaServer Faces 技术) 提供一个标准的 HTML 呈现工具包, 它由可呈现 HTML 标记的 renderer 组成。
replica (副本)	(n.)目录服务器上的后缀, 通过复制协议与一个或多个其他后缀相链接。
replica cycle (复制周期)	参见 replication cycle (复制周期)。
replica directory server (副本目录服务器)	(n.)接收全部或部分数据副本的目录。
replica group (副本组)	(n.)持有特定复制区域实例的服务器。一个服务器可以是多个副本组的一部分。
replication (复制)	(n.)同步分布在多个 Directory Server 上的数据并纠正更新冲突的过程。
replication agreement (复制协议)	(n.)一组存储在供应者服务器上的配置参数, 这些参数指明了要复制的后缀、数据推送到的使用方服务器、可以发生的 replication (复制) 次数、供应者用来绑定到使用方的 DN 和凭证, 以及连接所采用的安全方式。
replication base entry (复制基条目)	(n.)所复制区域之根的 DN。
replication cycle (复制周期)	(n.)在两个或更多副本间交换更新信息的时间间隔。复制周期在尝试向另一副本或另一组副本推送数据或从中拉取数据时开始, 在成功交换数据或遇到错误时结束。
replication session (复制会话)	(n.)作为 replication cycle (复制周期) 的一部分, 在 replica group (副本组) 中的两个服务器间为传递更新信息而建立的会话。
request object (请求对象)	(n.)含有页面和会话数据的对象, 这些数据由客户机产生, 作为输入参数传递给 servlet 或用 JavaServer Pages technology (JavaServer Pages 技术) 创建的页面。
request-response messaging (请求-响应消息传送)	(n.)一种 messaging (消息传送) 方法, 在接收到响应前将一直保持阻塞状态。
required attribute list (必需属性列表)	(n.)指定对象类所必需的属性列表。必需属性有前导关键字 MUST。

- required attributes (必需属性)** (n.)使用某特定对象类的条目中必须要有有的属性。另参见 **allowed attributes (允许的属性)**、**attribute (属性)**。
- resource (资源)** (1)(n.)网络中可由 URL 标识的任何项, 如 Web 页、文档或 FTP 目录。在非正式场合, 常常将资源称为文档。
(2)(n.)服务器可以访问并发送给请求客户机的任何 URL、目录或程序。
(3)(n.)提供到系统(如数据库服务器和消息传送系统)的连接的程序对象。
- resource adapter (资源适配器)** (n.)一种系统级的软件驱动程序, EJB 容器和应用程序客户端用它来连接到企业信息系统(enterprise information system, **EIS**)。资源适配器通常特定于某一 EIS。它以库的形式存在, 并在使用它的服务器或客户机地址空间中使用。将资源适配器插入到容器中。应用程序组件被部署到容器上, 随后使用客户机 API (由适配器提供) 或工具所生成的高级抽象来访问底层 EIS。资源适配器和 EJB 容器共同提供实现与 EIS 之间的连接的底层机制(事务、安全性和连接入池)。另参见 **connector (连接器)**。
- resource adapter module (资源适配器模块)** (n.)一个包含所有 Java 接口、类和本机库的可部署单元, 实现资源适配器以及资源适配器部署描述符。
- resource calendar (资源日历)** (n.)与资源(如会议室)或设备(如笔记本电脑或高射投影仪)关联的日历。
- resource description (资源描述)** (n.)通过 URL 与资源关联的属性值对列表。资源描述可由代理自动生成, 也可人工编写。一旦组装好资源描述系统信息库, 服务器便可通过资源描述消息以程序化方式将该系统信息库导出, 以供 Web 代理对资源描述进行搜索和检索。资源描述以 **SOIF** 格式存储。
- resource description message (资源描述消息)** (n.)一种机制, 用以搜索和检索可通过网络访问的资源的元数据(称为资源描述)。
- resource invocation (资源调用)** (n.)在节点上运行的资源类型实例。这是一个抽象概念, 表示已在节点上启动的资源。
- resource manager (资源管理器)** (n.)可访问一组共享资源。资源管理器参与事务管理器从外部控制和协调的事务。资源管理器通常与访问它的客户机运行在不同的地址空间中或不同的机器上。注: 在资源和事务管理上下文中提及企业信息系统(enterprise information system, **EIS**)时, 将该系统称为资源管理器。
- resource manager connection (资源管理器连接)** (n.)一种对象, 代表具有资源管理器的 **session (会话)**。
- resource manager connection factory (资源管理器连接工厂)** (n.)用于创建资源管理器连接的对象。
- resource offering (资源提供)** (n.)在“搜索服务”中, 资源提供定义了一段身份认证数据与提供其访问权的服务实例之间的关联。
- resource reference (资源引用)** (n.)部署描述符中的一种元素, 用于标识组件的资源编码名称。

response buffer (响应缓冲区)	(n.)Portal Server Mobile Access 服务器响应缓冲区将较大的响应存储为多个单独的较小响应，以使它们适合有限的设备缓冲区。
response object (响应对象)	(n.)引用调用客户机并为该客户机提供输出生成方法的对象。
restart (重新启动)	(v.)在不删除其状态信息的情况下启动 robot ，这样做可使 robot 以其先前停止时的同一状态开始运行。与 fresh start (全新启动) 相对。
restore (恢复)	(v.)将文件夹内容从备份设备复制到消息存储。另参见 back up (备份) 。
ResultSet object (ResultSet 对象)	(n.)实现 <code>java.sql.ResultSet</code> 接口的对象。 ResultSet 对象用于封装一组从数据库或其他表格数据源检索到的行。
resync interval (再同步时间间隔)	(n.)连接器检查 适用于 Windows 的身份认证服务 目录源的更改的频率。该周期性检查非常有用，且只需读取从上一次检查后发生更改的用户条目。控制台以毫秒为单位显示该值，默认值为 1000 (1 秒)。
retro changelog (追溯更改日志)	(n.)按更改到达本地服务器的顺序而不是其应用于系统的顺序存储这些更改。追溯更改日志并非设计用于多主复制环境。与 change log (更改日志) 不同，追溯更改日志并不在复制中使用。它提供与 Directory Server 4 的向后兼容性。
reusable component (可重用组件)	(n.)为了能在一个以上的功能体中使用（例如，被一个以上的资源或应用程序使用）而创建的组件。
reverse DNS lookup (反向 DNS 查找)	(n.)通过查询 DNS 将数值 IP address (IP 地址) 分解为相当的 gateway (网关) 的过程。
reverse proxy (反向代理)	(n.)在客户机与服务器之间执行双向 URL 重写和转换的代理。与存在于客户机端的代理不同，反向代理存在于网络的服务器端。在 Java Enterprise System Portal Server 中，反向代理存在于 Java Enterprise System Portal Server Secure Remote Access Pack 中。
Rewriter (重写器)	(n.)重写器提供了一个 Java 类库，用于以各种 Web 语言（如 HTML、Javascript 和 XML）和在 HTTP 位置头中（重定向）重写 URL 引用。重写器定义了 Java Enterprise System Directory Server Access Management Edition 服务来存储规则，这些规则定义要如何进行重写以及要重写的数据。重写器还包括一个用于编辑这些规则的管理控制台模块。
rewrite rule (重写规则)	(n.) MTA 用来将消息路由到正确主机以进行传送的工具。重写规则执行以下功能：(1) 从收到的消息的地址中提取主机和域说明；(2) 以重写规则模式对主机和域说明进行匹配；(3) 根据域模板重写主机和域说明；(4) 决定应将消息置于哪个通道队列。又称为域重写规则。
RFC	(request for comments, 请求注解) (n.)由 Internet 工程任务组维护的文档系列，描述 Internet 协议套件以及相关实验。几乎没有 RFC 描述 Internet 标准，但所有 Internet 标准都作为 RFC 发布。参见 http://www.imc.org/rfcs.html 。
RMI	(remote method invocation, 远程方法调用) (n.)一种技术，允许在某台 Java 虚拟机上运行的对象调用在另一台 Java 虚拟机上运行的对象。
RMI-IIOP	(n.)一种旨在使用 CORBA IIOP 协议的 RMI 版本。如果最初将所有 remote 接口都定义为 RMI 接口，则 RMI over IIOP 可与以任何语言实现的 CORBA 对象进行互操作。

- RMIC** (n.)远程方法调用编译器。
- robot** (n.)一个程序，用于查找位于网络特定部分的所有资源。
- robot application function (robot 应用函数)** 参见 [RAF](#)。
- role (角色)**
- (1) (n.)一种由应用程序组合者定义的抽象用户逻辑分组。部署应用程序时，在操作环境中将角色映射到安全身份认证（如用户（主体）或组）。另参见 [user \(用户\)](#)、[group \(组\)](#)。
 - (2) (n.)在 J2EE 服务器 [authentication \(验证\)](#) 服务中，指有权访问一组特定资源的抽象名称。
 - (3) (n.)在 Java Enterprise System Directory Server Access Management Edition 中，指的是表示特许操作选择集的分组。通过将角色应用于用户或服务，主体可以执行这些操作。例如，通过将某些权限限定于“员工”角色或“经理”角色并将该角色应用于某用户，该用户的访问能力便被限定于该角色所授予的权限范围内。角色是使用访问控制指令 ([access control instruction, ACI](#)) 定义的。
 - (4) (n.)一种由使用 J2EE 技术开发的应用程序的开发和部署阶段中的某一方执行的功能。角色包括：应用程序组件提供者、应用程序组合者、部署者、J2EE 产品提供者、EJB 容器提供者、EJB 服务器提供者、Web 容器提供者、Web 服务器提供者、工具提供者和系统管理员。
- role-based attributes (基于角色的属性)** (n.)因条目在关联的 [CoS](#) 模板内拥有某个特定角色而出现在该条目中的属性。
- role mapping (角色映射)** (n.)将容器所识别的组或主体（或二者）与在部署描述符中指定的安全角色关联起来的过程。在服务器中安装组件之前，部署者必须先映射安全角色。
- rollback (回滚)** (n.)撤销 [transaction \(事务\)](#)。[transaction \(事务\)](#) 中的某个时刻，此时取消事务中涉及的所有对资源的更新。
- root (超级用户或根)**
- (1) (n.)（仅限 UNIX）UNIX 机器上享有最高特权的用户。超级用户拥有对机器上所有文件的完全访问权限。
 - (2) (n.)XML 文档中最外层的元素。该元素包含所有其他元素。
- root DN (根 DN)** (n.)[Directory Manager \(目录管理员\)](#) 的 DN。
- Root DSE (根 DSE)** (n.)一个条目，由 Directory Server 自动生成，并以一个空（长为字节）的 DN 从 `baseObject` 搜索中返回。“根 DSE”向客户机提供有关服务器配置的信息，如指向 [subschema entry \(子模式条目\)](#) 的指针、服务器所持命名上下文的 DN 列表以及服务器支持的 [LDAPv3](#) 控件和扩展名的列表。另参见 [DSE](#)。
- root entry (根条目)** (n.)[DIT](#) 分层结构的顶级条目。
- root suffix (根后缀)** (n.)一个或多个 [sub suffix \(子后缀\)](#) 的父项。目录树可包含一个以上的根后缀。

router (路由器)	(n.)负责确定网络通信流动路径的系统。路由器使用路由选择协议获得有关网络和算法的信息，以根据称为“路由选择矩阵”的多个判别条件选择最佳路由。在“开放系统互连”术语中，路由器为“网络层”中间系统。另参见 gateway (网关) 。
routing (路由选择)	参见 message routing (消息路由选择) 。
routing tables (路由表)	(n.)存放有关消息始发者和接受者信息的内部数据库。
row (行)	(n.)单个数据记录，其中含有表中每一列的值。
RowSet object (RowSet 对象)	(n.)一个对象，其中封装了一组从数据库或其他表格数据源检索到的行。RowSet 对象扩展了 <code>java.sql.ResultSet</code> 接口，从而使 ResultSet object (ResultSet 对象) 能够充当基于 <code>JavaBeans™</code> 组件体系结构的组件。
RPC	(remote procedure call, 远程过程调用) (n.)一种用于访问远程对象或服务的机制。
RTT	(round trip time, 往返时间) (n.)在一个闭合回路（从服务器到客户机，再从客户机返回服务器）中传输一个信号所需的时间。这一延迟在要求双向交互通信的系统中很重要。在此系统中，RTT 会直接影响吞吐率。在 <code>Java Enterprise System Directory Server</code> 上下文中，RTT 和 TCP 窗口会通过广域网对复制性能产生显著影响。又称为 round-trip delay time (往返延迟时间) 。
rules (规则)	(n.)用于确定某个条件是否得到满足的逻辑测试。robot 将规则用作过滤器的组成部分，以确定要建立索引的内容类型，及在分类规则中确定分配给资源的类别。

S

- SAAJ** (SOAP with Attachments API for Java, 用于 Java 的带有附件的 SOAP API) (n.) SOAP 消息传送的基本软件包, SAAJ 包含用于创建和填充 SOAP 消息的 API。
- SAF** (server application function, 服务器应用函数) (n.) 参与处理请求和其他服务器活动的函数。
- safe file system (安全文件系统)** (n.) 一种文件系统, 可执行日志记录功能。这样, 当某个系统崩溃时, 该系统可以将数据回滚到崩溃前的状态, 并恢复所有数据。例如, Veritas 文件系统 (Veritas File System, VxFS) 就是一个安全文件系统。
- SASL** (simple authentication and security layer, 简单验证和安全层) (n.) 一种控制机制的方法, POP、IMAP 或 SMTP 客户机通过该机制向服务器表明身份。Java Enterprise System Messaging Server 对使用 SMTP SASL 的支持符合 RFC 2554 (ESMTP AUTH) 要求。SASL 在 RFC 2222 中定义。另参见 POP3 和 IMAP4。
- SAX** (Simple API for XML, XML 简单 API) (n.) 一种事件驱动接口, 发生解析事件时, parser (解析程序) 将通过它来调用由调用方提供的多种方法中的其中一个。事件包括识别 XML 标记、查找错误、引用外部实体或处理 DTD 规范。
- schema (模式)**
- (1) (n.) 说明何种信息类型可以在目录中存储为条目的定义。当与模式不匹配的信息存储在目录中时, 试图访问该目录的客户机可能无法显示正确结果。
 - (2) (n.) database (数据库) 中的表和列的结构。在 Application Server 中, 可从 entity bean (实体 bean) 自动生成模式。
 - (3) (n.) 一种数据库驱动的方法, 用于通过基于 XML 的语言指定 XML 文档的约束。模式处理 DTD 文件中的缺陷 (例如不能约束可出现在特定字段中的数据种类)。模式建立在 XML 的基础上, 它们具有层次结构。因此可更容易地创建明确的规范, 以及确定应用注释的范围。
- schema checking (模式检查)** (n.) 一种可确保在目录中添加或修改的条目与所定义的模式相符的检验过程。默认情况下, 模式检查处于启用状态。当用户试图保存与模式不符的条目时, 会收到一个错误提示。

schema name (模式名称)	(n.)SOIF 的模式或类型。例如, 文档的 SOIF 的模式名称为 @DOCUMENT, 而 resource description message (资源描述消息) 标头的模式名称为 @RDMHeader。
SCM	参见 service control manager (服务控制管理器) 。
scoping (作用范围)	(n.)在导入代理导入的资源说明中设置的限制条件。所用的语法与用户搜索的语法相同。
search base (搜索基)	参见 base DN (基 DN) 。
Search database (搜索数据库)	(n.)通常由 robot 生成的资源说明的可搜索数据库。另参见 robot 。
search data hiding rule (搜索数据隐藏规则)	(n.)一种规则, 用于确定 Directory Proxy Server 应如何过滤搜索操作的结果并将其返回给客户机。
Search Engine (搜索引擎)	(n.)集成在 Portal Server 6.0 中的搜索功能, 以前称为 Compass Server (Portal Server 3.0)。Search Server 持有由 robot 收集的资源说明组成的数据库, 这些资源说明通常都进行了分类。用户可以搜索资源说明或通过浏览类别查找特定资源。
secondary data view (从数据视图)	(n.)两个组成连接数据视图的 Directory Proxy Server 数据视图之一。从数据视图通常提供关于主数据视图中的条目的附加信息。另参见 primary data view (主数据视图) 。
secondary directory server (辅助目录服务器)	(n.)多主复制环境中的主目录服务器主代理实例, 在首选目录服务器不可用时, 适用于 Windows 的身份认证服务可使用此服务器。在首选目录服务器不可用时, 适用于 Windows 的身份认证服务可将在“活动目录”或 Windows NT 中所做的更改同步到辅助目录服务器, 但只有在首选目录服务器可用后, 才会同步在辅助服务器或任何其他目录服务器主系统中所做的更改。
Secure Remote Access (SRA)	(n.)SRA 允许大多数客户机设备通过安全连接访问个性化门户应用程序、内容、文件和服务。也称为 Sun Java™ System Portal Secure Remote Access (SRA)。
secure socket layer (安全套接字层)	参见 SSL 。
security (安全性)	(n.)一种确保应用程序资源仅由授权客户机访问的筛选机制。
security attribute (安全属性)	(n.)与 principal (主体) 相关的 attribute (属性) 。安全属性可通过 authentication (验证) 协议或 J2EE 产品提供者 (或二者) 与主体相关联。
security constraint (安全约束)	(n.)注释 Web 内容的预订保护的声明性方法。安全约束由 web resource collection (Web 资源集合) 、 authorization constraint (授权约束) 和 user data constraint (用户数据约束) 组成。
security context (安全上下文)	(n.)一种对象, 在两个实体之间封装与安全性相关的共享状态信息。
security-module database (安全模块数据库)	(n.)含有描述 SSL 加密器硬件加速器信息的文件。又称为 secmod 。
security permission (安全权限)	(n.)一种由 J2SE 定义的机制, J2EE 平台使用它来表示应用于应用程序组件开发者的编程限制。

security permission set (安全权限集)	(n.)J2EE 产品提供者要执行每个组件类型所必须提供的最小安全权限集。
security policy domain (安全策略域)	参见 realm (领域) 。
security role (安全角色)	参见 role (角色) 。
security technology domain (安全技术域)	(n.)一个范围, 在其中使用相同的安全机制来加强安全策略。在一个技术域中可存在多个安全策略域。
security view (安全视图)	(n.)一组由应用程序组合者定义的安全角色。
self access (自访问权限)	(n.)如果将该权限授予用户, 则表示当 bind DN (绑定 DN) 与目标条目匹配时, 用户对他们自己的条目具有访问权限。
self-generated certificate (自生成证书)	(n.)仅当用条目的公共值的消息摘要为条目进行命名并且这些名称进行安全通信时, 才能使用的公共密钥值。另参见 issued certificate (已颁发证书) 。
sendmail	(n.) (仅限 UNIX) 通用 MTA 。在大多数应用程序中, Java Enterprise System Messaging Server 可用作 sendmail 的替换。
serializable object (可序列化对象)	(n.)可以解构和重构、从而使之可在多个服务器之间存储和分发的对象。
server (服务器)	(n.)一种多线程软件进程 (有别于硬件服务器) , 可为通过外部接口访问服务的客户机提供一组分布式或紧密结合的服务。
server administrator (服务器管理员)	(n.)执行服务器管理任务的人员。服务器管理员根据任务 ACI 提供对特定服务器的任务的受限访问。配置管理员必须为用户分配对服务器的访问权限。当某个用户拥有服务器访问权限时, 该用户就是一个服务器管理员, 可以向用户提供服务器访问权限。
server assembly (服务器集合体)	(n.)一组相互依赖或密切相关到足以作为一个整体安装和部署的 Java Enterprise System 服务器。
server authentication (服务器验证)	(n.)一种可使客户机确保连接到安全服务器的验证方法, 可防止其他计算机仿冒服务器或避免将并不安全的服务器当作安全服务器。
server certificate (服务器证书)	(n.)与 HTTPS 协议搭配使用来验证 Web 应用程序。此证书可自签名, 也可由证书授权机构 (certificate authority, CA) 批准。只有安装了服务器证书之后, Application Server 的 HTTPS 服务才能够运行。
server daemon (服务器守护进程)	(n.)在运行时侦听并接受来自客户机的请求的进程。
server farm (服务器群)	(n.)在 Web Server 中, 服务器群是包含一个或多个运行不同配置的节点的网络。相反, cluster (群集) 是包含运行相同配置和 Web 应用程序的节点的网络。

- server instance (服务器实例)** (1) (n.)在同一计算机上安装的同一 Application Server 可以含有多个实例。每个实例都有自己的目录结构、配置和部署的应用程序。每个实例还可以含有多个虚拟服务器。另参见 [virtual server \(虚拟服务器 \)](#)。
- (2) (n.)Directory Server 或 Directory Proxy Server 实例。通过实例路径定义实例，而且该实例拥有相关数据库和配置文件。可在一个主机系统上运行多个实例。
- Server Message Block protocol (Server Message Block 协议)** (n.)一种协议，可为计算机中的客户机应用程序提供对文件的读取和写入权限，以及从计算机网络中的服务器程序请求服务的方法。SMB 协议可通过 Internet 在其“传输控制协议”或其他网络协议（例如 Internetwork Packet Exchange 和 NetBEUI）之上使用。Java Enterprise System Portal Server 对 NetFile 使用 SMB。
- server plug-in API (服务器插件 API)** (n.)一个扩展，允许扩展和自定义 Java Enterprise System 服务器的核心功能，并提供一个可伸缩的高效机制，用于构建 HTTP 服务器与后端应用程序之间的接口。又称为 NSAPI。
- server principal (服务器主体)** (n.)服务器以其身份执行的操作系统 [principal \(主体 \)](#)。
- server process (服务器进程)** (n.)一种由操作系统设置的自包含、功能完全的执行环境。应用程序的每个实例通常在一个单独的进程中运行。
- server root (服务器根目录)** (1) (n.)服务器计算机上的一个目录，专门用于存放服务器程序及配置文件、维护文件和信息文件。又称为 ServerRoot 或 [domain directory \(域目录 \)](#)。
- (2) (n.)相对于服务器上其他文件的一个目录位置。例如，Solaris 系统的默认 Calendar Server 安装使用路径 /opt/SUNWics5/ 作为服务器根目录。
- (3) (n.)用于安装所有与特定主机上的特定 Java Enterprise System Administration Server 关联的 Java Enterprise System 服务器的目录。另参见 [installation directory \(安装目录 \)](#) 和 [instance directory \(实例目录 \)](#)。
- server-side rules (服务器端规则)** (n.)一组实现服务器端邮件过滤的规则。基于 Sieve 邮件过滤语言。
- service (服务)** (1) (n.)服务器提供的一种功能。例如，Java Enterprise System Messaging Server 提供 SMTP、POP、IMAP 和 HTTP 服务。
- (2) (n.)针对一个或多个客户机执行的一种软件功能。此功能可能是非常低层级的（如内存管理），也可能是高级的（如信用核查 [business service \(业务服务 \)](#)）。高级服务可以由一族单项服务组成。服务可以是本地的（供本地客户机使用），也可以是分布式的（供远程客户机使用）。
- service control manager (服务控制管理器)** (n.)（仅限 Windows NT）一种用于管理服务的管理程序。
- service element (服务元素)** (n.)一个或多个 [connector \(连接器 \)](#) 组件的组合，这些组件共享一个引擎组件来处理收到的请求。
- service endpoint interface (服务端点接口)** (n.)一种 Java 接口，声明客户机可在 [web service \(Web 服务 \)](#) 上调用的方法。

service quality component (服务质量组件)	(n.)Java Enterprise System 包括的多种 system component (系统组件) 之一。支持组件 (包括访问组件和管理组件) 为 system service (系统服务) 提供支持。
service-oriented architecture (面向服务的体系结构)	参见 SOA。
service provider (服务提供者)	(n.)提供基于 Web 的服务的商业组织或非赢利性组织。可以包括 internet 门户、零售商、运输提供者、金融机构、娱乐公司、图书馆、大学和政府机构。
Service Registry	(n.)包括在 Java Enterprise System 中的 ebXML registry (ebXML 注册表) 产品。
service stack (服务栈)	(n.)支持分布式企业应用程序所需的分布式服务的分层。该分层反映栈中较高级别服务对其下层服务的依赖性。
servlet	(1) (n.)使用扩展 Web 服务器功能的 Java 编程语言编写的服务器端程序, 利用请求-响应范例生成动态内容, 并与 Web 应用程序进行交互。Servlet 在服务器端运行, 这一点与 applet 相似, 但 servlet 不使用用户界面。 (2) (n.)Servlet 类的一个实例。一个 servlet 就是在服务器上运行的一个可重用应用程序。在 Java Enterprise System Application Server 中, servlet 通过执行表示逻辑、调用业务逻辑及调用或执行表示布局, 充当应用程序中每次交互的中央分发程序。
servlet container (servlet 容器)	(n.)一个提供网络服务的 container (容器) , 通过它来发送请求和响应、解码请求以及格式化响应。所有 servlet 容器必须支持 HTTP 作为请求和响应协议, 并且还支持其他请求-响应协议 (如 HTTPS)。
servlet container, distributed (可分布的 servlet 容器)	(n.)一种 servlet 容器, 可运行标记为可分布的 Web 应用程序, 且可跨在同一主机或不同主机上运行的多个 Java 虚拟机执行。
servlet context (servlet 上下文)	(n.)一个对象, 包含运行 servlet 的 Web 应用程序的 servlet 视图。servlet 可通过上下文来记录事件、获取资源的 URL 引用、设置和存储同一上下文中的其他 servlet 可使用的属性。
servlet engine (servlet 引擎)	(n.)处理所有 servlet 元功能的内部对象。总体来说, 即为 servlet 提供服务的一组进程, 包括实例化和执行。
servlet mapping (servlet 映射)	(n.)定义 URL 模式与 servlet 之间的关联。使用映射来将请求映射到 servlet。
servlet runner (servlet 运行器)	(n.)servlet 引擎的组成部分, 可调用含有一个请求对象和一个响应对象的 servlet。参见 session bean (会话 bean) 。
session (会话)	(1) (n.)servlet 或有状态会话 bean 所使用的一种对象, 用于跟踪多个 HTTP 请求中用户与 J2EE 或 Web 应用程序之间的交互。另参见 persistence (持久性) 。 (2) (n.)客户机服务器连接的一个实例。另参见 client-server model (客户机/服务器模型) 。 (3) (n.)对于 Java Enterprise System Portal Server, 是指用户与一个或多个应用程序之间的交互序列, 从登录开始, 到注销或超时结束。

(4)(n.)对于 Message Queue, 是指用于发送和接收消息的单线程上下文。它可以是一个队列会话或一个主题会话。

session bean (会话 bean)

(n.)由客户机创建, 且通常仅在单个客户机服务器会话期间存在的企业 bean。会话 bean 为客户机执行操作, 如计算或访问其他企业 bean。虽然会话 bean 可以是事务性的, 但它无法在系统发生崩溃时恢复。会话 bean 对象可以是无状态的 (不与特定客户机相关联), 也可以是有状态的 (与特定客户机相关联), 因此它们可以跨多个方法和事务维护对话状态。另参见 [stateful session bean \(有状态会话 bean\)](#)。

session cookie (会话 cookie)

(n.)返回给含有用户会话标识符的客户机的 cookie。另参见 [sticky cookie \(粘性 cookie\)](#)。

session failover (会话故障转移)

(n.)Access Manager 中的一种故障转移, 将 Sun Java System Message Queue 用作通信代理, 将 Berkeley DB 用作会话存储数据库。它不使用任何 Web 容器会话管理工具。如果单个硬件或软件出现故障, Access Manager 会话故障转移会保留用户的已验证会话状态, 从而将用户会话故障转移到辅助 Access Manager 实例, 而不会丢失任何会话信息, 也无需用户重新登录。另参见 [failover \(故障转移\)](#)、[persistence \(持久性\)](#)。

session key (会话密钥)

(n.)一种常见的加密技术, 使用单独的密钥对两人之间的每个对话进行加密。

session timeout (会话超时)

(n.)一个指定的时间段, 服务器可以在此时间段后使用户会话失效。

SGML

(Standard Generalized Markup Language, 标准通用标记语言) (n.)HTML 和 XML 的父项。虽然 HTML 共享 SGML 的特性来将表示信息嵌入到标记中, 但是 XML 是一种标准, 它允许完全地将信息内容从机制中分离出来以转译该内容。

shared component (共享组件)

(n.)Java Enterprise System 包括的多种 [system component \(系统组件\)](#) 之一。共享组件 (通常是程序库) 为其他系统组件提供本地服务。相反, [system service \(系统服务\)](#) 向其他系统组件 (或向 [application component \(应用程序组件\)](#)) 提供分布式基础结构服务。

shared component descriptor file (共享组件描述符文件)

(n.)含有指定共享组件的元数据的文件 (通常为 XML 格式)。

shared folder (共享文件夹)

(n.)可由一个以上人员读取的文件夹。共享文件夹有一个所有者, 他可以指定该文件夹的读取权限并可从共享文件夹中删除消息。共享文件夹还可以有一个仲裁人, 他可以编辑、拦截或转发收到的消息。只有 IMAP 文件夹可以共享。另参见 [personal folder \(个人文件夹\)](#)、[public folder \(公共文件夹\)](#)。

shared-key cryptography (共享密钥密码学)

(n.)一种各方必须使用相同的密钥加密或解密密文的密码学。又称为对称密钥密码学。

SHTML

(server-side include Hypertext markup language, 服务器端包含超文本标记语言) (n.)一种 HTML 文件, 其中包括嵌入式服务器端包含 (server-side include, SSI)。

Sieve

(n.)建议的邮件过滤语言。

Simple API for XML (XML 简单 API)

参见 [SAX](#)。

simple authentication and security layer (简单验证和安全层)	参见 SASL 。
simple index (简单索引)	(n.)一种目录列表, 仅显示文件名称, 不带任何图形元素。与复杂索引相对。
Simple Mail Transfer Protocol (简单邮件传输协议)	参见 SMTP 。
Simple Network Management Protocol (简单网络管理协议)	参见 SNMP 。
Simple Object Access Protocol (简单对象访问协议)	参见 SOAP 。
SIMS	(n.)Solstice Internet Mail Server™ 和 Sun Internet Mail Server™。
single field substitution string (单个字段替换字符串)	(n.)在重写规则中, 可动态重写主机的指定地址令牌和域地址的域模板的组成部分。另参见 domain template (域模板)。
single identity (单一身份)	(n.)用户凭借 Java Enterprise System 目录中的单个用户条目所具有的身份。基于此单个用户条目, 用户可以获准访问各种 Java Enterprise System 资源, 如门户、Web 页以及诸如消息传送、日历和即时消息传送等服务。
single logout (单点注销)	(n.)用户从身份认证提供者或服务提供者注销, 以及从该验证域的所有服务提供者或身份认证提供者注销的能力。
single sign-on, SSO (单点登录)	(1) (n.)一种功能, 允许将通过了分布式系统中一项服务的用户验证自动应用于系统中的其他服务。 (2) (n.)一种情况, 在此情况下用户的验证状态可跨单个虚拟服务器实例中的多个 J2EE 应用程序共享。参见 SSO 。 (3) (n.)当拥有联合身份的用户向身份认证提供者验证时建立的验证进程。由于该用户拥有联合身份, 该用户不必重新验证即可访问附属的服务提供者。
site (站点)	(n.) robot 为查找资源而访问的网络位置。在 site definition (站点定义) 中确定站点的地址以及要在该站点索引的文档的种类。
site configuration (站点配置)	(n.)一种提供简化配置的能力, 以允许 Access Manager 客户机与多个负载平衡的 Access Manager 实例进行通信。站点配置支持在每个站点部署多个负载平衡器和防火墙。
site definition (站点定义)	(n.)关于 robot 为寻找资源可以访问的位置的约束。可以使用站点定义将 robot 限定到一个特定的服务器、一个特定的服务器组或一个域。站点定义包括描述 robot 应从站点索引的文档类型的过滤器。

SIZE	(n.)一个 SMTP 扩展，可以使客户机向服务器声明某一特定消息的大小。服务器可以基于所声明的消息大小，告诉客户机其是否愿意接受该消息。服务器可以向客户机声明它愿意接受的最大消息大小。在 RFC 1870 中定义。
slapd.exe	(n.) (仅限 Windows) 负责 Directory Server 所有操作的进程或服务。在 UNIX 系统中，与之对应的是 ns-slapd 。
slave channel program (从通道程序)	(n.)接受由远程系统启动的传输的通道程序。另参见 master channel program (主通道程序)。
smart host (智能主机)	(n.)某个域中的邮件服务器，当其他邮件服务器无法识别收件人时，将消息转发到该邮件服务器。
SMB protocol (SMB 协议)	参见 Server Message Block protocol (Server Message Block 协议)。
SMTP	(Simple Mail Transfer Protocol, 简单邮件传输协议) (n.)Internet 中最常用并受 Java Enterprise System Messaging Server 支持的电子邮件协议。该协议在 RFC 821 中定义，相关联的消息格式说明包含在 RFC 822 中。
SMTP AUTH	参见 AUTH 。
SMTP proxy (SMTP 代理)	(n.)SMTP 的变体，将消息从网络中的一台计算机发送到另一台计算机，并在 Internet 上用于对电子邮件进行路由。
sn attribute (sn 属性)	(n.)姓氏的 LDAP 别名。
SNMP	(Simple Network Management Protocol, 简单网络管理协议) (n.)用于交换有关网络活动的数据的协议。通过 SNMP，数据可在受管理的设备 (任何运行 SNMP 的设备，如主机、路由器、您的 Web 服务器，以及您网络上的其他服务器) 和 NMS 之间传输。
SNMP master agent (SNMP 主代理)	(n.)在各种子代理和 NMS 之间交换信息的软件。
SNMP SOCKS	(n.)一种防火墙软件，当受到防火墙软件或硬件 (例如，路由器配置) 阻止而不能建立直接连接时，该防火墙软件可建立防火墙内部到外部的连接。
SNMP subagent (SNMP 子代理)	(n.)收集有关受管理设备的信息，并将该信息传递给主代理的软件。
SOA	(service-oriented architecture, 面向服务的体系结构) (n.)描述由服务的使用方和提供者组成的复合应用程序。使用方和提供者无需引用对方的具体位置即可交换消息。此体系结构还将应用程序的核心进程与其他提供者和使用方隔离开来。
SOAP	(Simple Object Access Protocol, 简单对象访问协议) (n.)用于在一个分散的分布式环境中交换结构化信息的轻量协议。它使用 XML 技术来定义可扩展 essaging (消息传送) 框架，该框架包含可通过多种底层协议交换的消息结构。
SOAP with Attachments API for Java (用于 Java 的带有附件的 SOAP API)	参见 SAAJ 。

soft restart (软重启)	(n.)一种重新启动服务器的方式, 可使服务器通过重新读取其配置文件进行内部重启。软重启向进程发送 HUP 信号(信号 1)。该进程本身不会像在硬重启中那样彻底终止。
SOIF	(summary object interchange format, 摘要对象交换格式) (n.)用于传送资源说明及其他种类的结构化对象的语法。每个资源说明表示为一个属性值对列表。SOIF 将文本和二进制数据作为值和带一些微小扩展的多值属性处理。SOIF 是一种允许许多资源说明以单个高效的流大量传输的流化格式。
SOIF attribute (SOIF 属性)	(n.)一种数据库属性类型。搜索数据库中的每个资源说明都拥有多个属性或字段。这些属性称为 SOIF 属性。
Solaris™ logical name (Solaris 逻辑名称)	(n.)该名称通常用于管理 Solaris 操作系统设备。对于磁盘, 它们通常具有类似 /dev/rdisk/c0t2d0s2 的形式。对于每个 Solaris 逻辑设备名称, 都存在一个基础 Solaris 物理设备名称。另参见 Solaris physical name (Solaris 物理名称) 。
Solaris physical name (Solaris 物理名称)	(n.)在 Solaris 操作系统中由设备驱动程序为设备指定的名称。该名称在 Solaris 计算机上显示为 /devices 树下的一个路径。例如, 一个典型 SI 磁盘拥有一个类似下面形式的 Solaris 物理名称: devices/sbus@1f,0/SUNW,fas@e,8800000/sd@6,0:c,raw。另参见 Solaris logical name (Solaris 逻辑名称) 。
solution life cycle (解决方案生命周期)	(n.)规划和跟踪部署项目的工具。生命周期构建成功地将规划部署到一系列有序的阶段所必需的准备、分析和设计。每个阶段都由相关任务组成, 这些任务产生的输出将作为后续阶段的输入。每个阶段内的任务都是反复进行的, 生成每个阶段的输出之前要求进行彻底的分析和设计。
spare node (备用节点)	(n.)可替换发生故障的活动节点的 HADB 节点。如果活动节点出现故障, 备用节点会从镜像节点复制数据并转为活动状态。另参见 HADB node (HADB 节点) 、 active node (活动节点) 、 mirror node (镜像节点) 和 data redundancy unit (数据冗余单元) 。
spider	参见 robot 。
spoofing (网络欺骗)	(n.)一种试图对服务器进行访问或向其发送消息的客户机虚报其主机名的网络攻击形式。
SQL	(structured query language, 结构化查询语言) (n.)定义数据库对象和处理数据的标准化关系数据库语言。SQL2 和 SQL3 指明了该语言的版本。
SQL/J	(n.)一组标准, 包括在 Java 编程语言中将 SQL 语句嵌入到方法中的规范, 以及将 Java 静态方法调用为 SQL 存储过程和用户自定义函数的规范。SQL 检查程序可在程序开发期间检查静态 SQL 语句中的错误, 而不像使用 JDBC 驱动程序时在执行时检查错误。
SSL	(secure socket layer, 安全套接字层) (n.)一种安全、低层加密形式, 由其他协议(如 HTTP 和 FTP)使用。SSL 协议包括服务器验证、传输数据加密以及可选客户机验证的置备。此协议可使客户机-服务器应用程序在进行通信时不会被窃听或篡改。
SSL authentication (SSL 验证)	(n.)一种验证方法, 该方法通过将客户机证书中的信息作为身份证明或通过检验 LDAP 目录中发布的客户机证书来使用安全性证书确认用户的身份。
SSL certificate (SSL 证书)	(n.)一种表示您或供应商已批准使用 PKI 对安全事务进行加密和解密的电子令牌。您在安装 Java Enterprise System Portal Server 软件时将创建一个自签名的“SSL 证书”。不过, 您也可以通过 Internet 从授权使用安全通信服务的证书供应商处获得“SSL 证书”。
SSO	参见 single sign-on, SSO (单点登录) 。

SSR	参见 server root （服务器根目录）。
standard index （标准索引）	(n.)在默认情况下维护的索引。
starting points （起点）	(n.)搜索引擎 robot 为开始枚举资源而访问的站点列表。
state （状态）	(1) (n.)实体在任意给定时间的情况或状态。 (2) (n.)一种分布式数据存储机制，可用于存储使用 Java Enterprise System Application Server 功能接口 <code>IState2</code> 的应用程序的状态。另参见 conversational state （会话状态）、 persistent state （持久状态）。
stateful session bean （有状态会话 bean）	(n.)一种会话 bean，代表与特定客户机进行的会话，并跨多个客户机调用的方法自动维护会话状态。
stateless session bean （无状态会话 bean）	(n.)一个表示无状态服务的会话 bean。无状态会话 bean 是完全瞬态的，并封装了特定客户机在有限时间跨度内所需的临时业务逻辑片段。无状态会话 bean 的所有实例都是相同的。
static group （静态组）	(n.)通过枚举每个组成员静态定义的邮件组。另参见 dynamic group （动态组）。
static web content （静态 Web 内容）	(n.)静态 HTML 文件、图像、applet Java 归档 (Java archive, JAR) 文件，以及其他任何可由 Web 服务器在不使用 Java Web 容器的情况下直接提供的内容。对于 Java Enterprise System Portal Server，Web 文件安装在 Web 服务器中（与动态 Web 应用程序的位置相同）。
status event （状态事件）	(n.)用户的状态，包括是否联机。
sticky cookie （粘性 cookie）	(n.)返回客户机并强制该客户机始终连接到相同服务器进程的 cookie 。另参见 session cookie （会话 cookie）。
sticky load balancing （粘性负载均衡）	(n.)一种 load balancing （负载均衡）方法，在该方法中，初始客户机请求已达到负载均衡，但后续请求被引导至与初始请求相同的进程。
stop word （停用词）	(n.)搜索功能中将其标识为不应搜索的词，例如“the”、“a”、“an”和“and”。又称为丢弃词。
stored procedure （存储过程）	(n.)在 SQL 中编写并在数据库中存储的语句块。可以使用存储过程执行任何类型的数据库操作，如修改记录、插入记录或删除记录。使用存储过程可藉由减少通过网络发送的信息量提高数据库性能。
streaming （流化处理）	(n.)管理数据如何通过 HTTP 通信的技术。当对结果进行流化处理后，数据的第一部分可立即使用。当结果未流化处理时，在使用结果的任何部分以前必须先接收整个结果。流化处理提供了一种可使大数据量以更高效率返回的方式，从而提升了应用程序的性能体验。
strftime function （ strftime 函数）	(n.)将日期和时间转换为字符串的函数。服务器在附加尾缀时使用此函数。对于日期和时间， strftime 函数有一种特殊的格式语言，服务器可在尾缀中使用这种格式语言以说明文件的上次修改日期。
subagent （子代理）	参见 SNMP subagent （ SNMP 子代理 ）。
subdomain （子域）	(n.) gateway （ 网关 ）的倒数第二部分，用于标识拥有该域名的公司或组织内的分公司或部门（例如， support.example.com 和 sales.example.com ）。子域并不是始终需要指定的。

subnet (子网)	(n.)IP address (IP 地址) 的一部分, 用于标识主机 ID 字块。
subordinate reference (次层参考)	(n.)目录服务器所持有的命名上下文的子命名上下文。另参见 knowledge information (知识信息)。
suborganization (子组织)	(n.)在 Java Enterprise System Directory Server Access Management Edition 中, 指在某个组织下创建的对象, 它为企业提供了对其部门和资源的更高控制粒度。例如, 在设置 Java Enterprise System Portal Server 时, 可能会在顶级对象 <code>isp</code> 下创建一个名为 <code>mycompany</code> 的子组织。
subschema entry (子模式条目)	(n.)此类条目包含部分目录树中的条目所使用的所有模式定义 (对象类、属性、匹配规则等项的定义)。
substring index (子串索引)	(n.)一种用于搜索的文件信息语言执行例程, 借此可对条目中的子串执行高效搜索。子串索引的每个索引键最限定为三个字符。
sub suffix (子后缀)	(n.)根后缀下的分支。
suffix (后缀)	(n.)目录树中其下存有数据的条目的名称。同一目录中可以存在多个后缀。每个数据库只有一个后缀。
summary object interchange format (摘要对象交换格式)	参见 SOIF 。
SunTM Cluster software (Sun Cluster 软件)	Sun Cluster 软件系统, 用于创建高度可用和可伸缩的服务。
Sun Java System Application Server	参见 Application Server 。
Sun Java System Communications Express	参见 Communications Express 。
Sun Java System Compass Server	参见 Compass Server 。
Sun Java System Connector for Microsoft Outlook	参见 Connector for Microsoft Outlook 。
Sun Java System Delegated Administrator	参见 Delegated Administrator 。
Sun Java System Directory Server	参见 Directory Server 。
Sun Java System Instant Messaging Client	参见 Instant Messaging Client (Instant Messaging 客户机)。

Sun Java System Message Queue	参见 Message Queue 。
Sun Java System Portal Secure Remote Access (SRA)	参见 Secure Remote Access (SRA) 。
Java System Portal Server	参见 Portal Server 。
Sun Java System Synchronization	(n.)一种软件，在 Microsoft Windows 个人计算机上运行，用户可利用它来使日历事件和任务与移动设备和个人信息管理器 (personal information manager, PIM) (如 Microsoft Outlook) 保持同步。
Sun Java System Web Server	参见 Web Server 。
supplier (供应者)	(n.)一种服务器，其中含有复制到使用方服务器的目录树或子树的原版。
supplier replica (供应者副本)	(n.)包含原版目录信息并且可以更新的副本。服务器可持有任意数目的主副本。
supplier directory server (供应者目录服务器)	(n.)任何将更改发送给其他目录服务器的目录服务器。另参见 consumer directory server (使用方目录服务器) 。
symlinks (符号链接)	(n.) (仅限 UNIX) 一种特殊的文件或目录，它指向另一文件或目录，从而使两个文件或目录具有相同的内容。
symmetric encryption (对称加密)	(n.)加密和解密使用同一密钥的加密方法。例如，“数据加密标准” (Data Encryption Standard, DES) 就是一种对称加密算法。
symmetric key cryptography (对称密钥密码学)	参见 shared-key cryptography (共享密钥密码学) 。
synchronization (同步)	(1) (n.)由主目录服务器对副本目录服务器进行的数据更新。 (2) (n.) MTA 目录高速缓存的更新。
Synchronization User List (同步用户列表)	(n.)定义要同步的 Sun 和 Windows 目录中的用户。“同步用户列表”可根据 LDAP 基 DN 或过滤器来限制要同步的用户范围。
system component (系统组件)	(n.)包括在 Java Enterprise System 中并通过 Java Enterprise System 安装程序进行安装的任何软件包或任何一组软件包。有多种系统组件：提供分布式基础结构 service (服务) 的 server (服务器) 、通过提供访问和管理服务来支持系统服务组件的 system service (系统服务) ，以及向其他系统组件提供本地服务的 shared component (共享组件) 。
system index (系统索引)	(n.)不能删除或修改的索引，因为它对 Directory Server 操作是至关重要的。

- system service (系统服务)** (n.)定义 Java Enterprise System 所提供的独特功能的一项或多项分布式 **service (服务)**。系统服务通常需要若干 **supplier (供应者)** 和/或若干 **shared component (共享组件)** 的支持。
- system service component (系统服务组件)** (n.)Java Enterprise System 包括的多种 **system component (系统组件)** 之一。系统服务组件提供主要的 Java Enterprise System 基础结构服务：门户服务、通信和协作服务、身份识别和安全服务、Web 和应用程序服务以及可用性服务。

T

tag (标记)	(n.)在 XML 文档中, 标记是一段描述数据单元或元素的文本。由于标记处于一对尖括号 (< 和 >) 之间, 因此可将其辨识为与数据不同的标签。要将此类标签语法当作数据, 可使用实体引用或 CDATA 部分。
takeover (接管)	参见 failover (故障转移) 。
target (目标)	(1) (n.)在访问控制上下文中, 目标指明了具体 ACI 应用于的目录信息。 (2) (n.)在 Application Server, 目标是应用程序部署或配置更改应用于的服务器实例。 (3) (n.)在 Apache Ant 中, 目标是一组要执行的任务。另参见 asant 、 build file (生成文件) 。
target entries (目标条目)	(n.)CoS 范围内的条目。
task (任务)	(n.)在客户端的 Calendar Express 中, 指定待做事情日历组成部分。在服务器端, 任务又称为 todo (待办事项) 。
taxonomy (分类法)	(n.)Java Enterprise System Portal Server 搜索引擎中资源的分类系统。
telnet proxy (远程登录代理)	(n.)位于远程登录客户机和远程登录服务器之间起智能中继作用的应用程序。
template (模板)	(n.)一组应用到由 XPath 表达式选择的节点的格式化指令。
template entry (模板条目)	参见 cooperating server (协作服务器) 。
timeout (超时设置)	(n.)一段指定的时间, 过了这段时间之后服务器将放弃尝试完成似乎已挂起的服务例程。
time zone (时区)	(n.)使用相同时间的地理区域。从 -12 至 +12 (GMT 为 0) 共有 25 个小时时区。每个时区以 GMT 为基准测定。多数时区都有由三个字母组成的本地化缩写名称。Calendar Server 还使用时区 ID (time zone ID, TZID) (如美国/洛杉矶或亚洲/加尔各答) 来标识时区。

TLS	(Transport Layer Security, 传输层安全) (n.)一种协议, 在传输层提供加密和认证, 这样, 无需对客户机和服务器应用程序进行较大更改, 数据流便能通过安全通道。它是 SSL 的标准, 一种基于公共密钥的协议。
todo (待办事项)	(n.)在服务器端, 指定待做事情的日历组成部分。在客户机端的 Calendar Express 中, 待办事项称为 task (任务) 。
tool provider (工具提供商)	(n.)提供用于开发、打包和部署 J2EE 应用程序的工具的组织或软件供应商。
top	(n.) (仅限 UNIX) 一些 UNIX 系统中用于显示系统资源当前使用状况的程序。
topic (主题)	(n.)管理员为实现 publish and subscribe delivery model (发布和订阅传送模型) 所创建的对象。可将主题视为内容分层结构中负责收集和分发针对该主题消息的节点。以主题作为中介, 便将消息发布者与消息订户分隔开来。
top-level administrator (顶级管理员)	(n.)拥有以下管理权限的用户: 可通过用于消息传送和协作的 Delegated Administrator GUI 或 CLI, 在整个 Messaging Server 名称空间中创建、修改和删除邮件用户、邮件列表、家庭帐户和域。默认情况下, 此用户可以担任拓扑中所有消息传递服务器的消息存储管理员。
top-level domain authority (顶级域授权)	(n.)主机名称分类的最高类别, 通常表示域的组织类型 (例如, .com 代表公司, .edu 代表教育机构) 或域的原属国家/地区 (例如, .us 代表美国, .jp 代表日本, .au 代表澳大利亚, .fi 代表芬兰)。
topology (拓扑)	(1) (n.)目录树在物理服务器间的划分方式以及这些服务器相互链接的方式。 (2) (n.)对机器、Application Server 实例和 HADB 节点, 以及它们之间的通信流的安排。参见 server instance (服务器实例) 、 HADB node (HADB 节点) 。
transaction (事务)	(1) (n.)一组数据库命令, 这些命令的成功或失败都密不可分。涉及的所有命令必须都成功, 整个事务才算成功。 (2) (n.)修改数据的原子工作单元。事务封装一条或多条程序语句, 所有程序语句都以 commit (提交) 或 rollback (回滚) 结束。事务使多个用户可同时访问同一数据存储库。
transaction attribute (事务属性)	(n.)在 enterprise bean (企业 bean) 的部署描述符中指定的值, 在调用企业 bean 的方法时, EJB 容器使用它来控制事务范围。事务属性可为下列值: Required、RequiresNew、Supports、NotSupported、Mandatory 或 Never。
transaction context (事务上下文)	(n.)事务的范围 (本地或全局)。参见 transaction context (事务上下文) 。
transaction isolation level (事务隔离层)	(n.)某个事务所修改数据的中间状态对于其他并发事务的可见程度, 以及其他事务所修改的数据的中间状态对于此事务的可见程度。
transaction manager (事务管理器)	(n.)提供支持事务划分、事务资源管理、同步和事务上下文传播所需的服务和管理功能。通常使用 XA protocol (XA 协议) 。另参见 global transaction (全局事务) 。
transaction recovery (事务恢复)	(n.)分布式事务的自动或手动恢复。
transience (瞬态)	(n.)一种协议, 当资源不用时将其释放。与 persistence (持久性) 相对。

transient failure (瞬态故障)	(n.)消息处理过程中出现的错误情况。远程 MTA 不能在消息送达时处理消息，但也许能在以后处理该消息。本地 MTA 将消息送回到队列，并安排在以后某一时间重新传输该消息。
Transport Layer Security (传输层安全)	(TLS)(n.)SSL的标准化形式。另参见 secure socket layer (安全套接字层)。
transport protocols (传输协议)	(n.)用于在两个 MTA 之间提供消息传送手段的协议，例如 SMTP 和 X.400 。
trust database (信任数据库)	(n.)含有公共和专用密钥的安全性文件。又称为 key-pair file (密钥对文件)。
trusted provider (受信提供者)	(n.) circle of trust (信任环) 中的一组服务提供者和身份认证提供者之一。用户可以在安全的环境中与受信提供者进行事务处理和通信。

U

UAProf	(n.)由“开放移动联盟”定义的规范，允许移动设备将其功能传达给网络服务器。
UBE	参见 unsolicited bulk email （未经许可的海量电子邮件）。
UDDI	（Universal Description, Discovery, and Integration，通用描述、搜索和集成）(n.)为搜索和集成提供全球 Web 服务注册。此为行业首创，旨在创建独立于平台的开放式框架以用于描述服务、搜索业务和利用 Internet 来集成业务服务以及注册。它正由一个供应商联盟开发。
Unicode	(n.)一种由 ISO 10646 和 Unicode 协会定义的 16 位字符集，该字符集将数字映射为世界范围内各种语言的字符。由于 16 位涵盖了 32,768 个代码，Unicode 大到足以包含世界上所有的语言，针对每个概念有不同字符的表意语言（如中文）除外。Java 编程环境中的所有源代码都以 Unicode 来编写。有关详细信息，参见 http://www.unicode.org/ 。
unified messaging （统一消息传送）	(n.)此概念是指对电子邮件、语音邮件、传真和其他通信形式使用单一消息存储。Java Enterprise System Messaging Server 是实现完全统一消息传递解决方案的基础。
uniform resource indicator （统一资源指示符）	参见 URI 。
uninstallation （卸载）	(n.)彻底删除软件组件的过程。
universal principal name （通用主要名称）	(n.)登录用户的值，包括并在一起的登录名及该用户所属的域。例如，域 <code>example.com</code> 中用户 <code>bill</code> 的“通用主要名称”是 <code>bill@example.com</code> 。又称为 UPN。
Universal Standard Products and Services Classification, UNSPSC （通用标准产品和服务分类）	(n.)对商品进行分类和标识的 schema （模式）。它用于销售端和购买端目录，并在分析支出中作为标准化帐户代码使用。
unparsed entity （未解析实体）	(n.)包含非 XML 内容的一般实体。根据其性质，未解析实体包含二进制数据。

- unsolicited bulk email (未经许可的海量电子邮件)** (n.)大批散发者通常出于商业目的发来的未请求和不需要的电子邮件。又称为垃圾邮件。
- upper reference (上层参考)** (n.)在 [DIT](#) 中, 指的是所持命名上下文高于您所在目录服务器命名上下文的目录服务器。
- URI** (uniform resource identifier, 统一资源指示符) (n.)抽象或物理资源的全局唯一标识符。URL 即为一种 URI, 它指定资源的检索协议 (用于 Web 应用程序的 [http](#) 或 [https](#)) 以及物理位置 (主机名和主机相对路径)。
- URL database repair (URL 数据库修复)** (n.)对已损坏的 URL 数据库进行修复和更新的过程, 损坏的原因可能是: 软件故障、系统崩溃、磁盘故障或文件系统已满。
- URL mapping (URL 映射)** (n.)文档目录物理路径名到用户定义别名的映射过程, 这样做使得目录中的文件只需引用目录的别名, 而无需引用文件的完整物理路径名。您不用将文件指明为 `usr/JES/servers/docs/index.html`, 而可将该文件指明为 `/myDocs/index.html`。此映射使得用户不必知道服务器文件的物理位置, 从而增强了服务器的安全性。
- URL path (URL 路径)** (n.)URL 的一部分, 由 HTTP 请求所传递以调用 servlet。URL 路由由上下文路径、servlet 路径以及路径信息组成, 如下:
- 上下文路径是与 servlet 上下文 (servlet 是它的一部分) 相关联的路径前缀。如果该上下文是位于 Web 服务器 URL 名称空间根部的默认上下文, 则路径前缀为空字符串。否则, 路径前缀以 / 字符开头, 但不以 / 字符结尾。
 - servlet 路径是直接与该请求的映射相对应的路径部分。该路径以 / 字符开头。
 - 路径信息是请求路径的一部分, 而该请求路径不是上下文路径或 servlet 路径的组成部分。
- URL pool (URL 池)** (n.)robot 要处理的 URL 列表。当 robot 启动时, URL 池由起始点组成, 但该池会随着枚举过程中发现的资源而迅速增大。
- URN** (uniform resource name, 统一资源名称) (n.)一种唯一标识符, 它标识实体但不指出其所在位置。尝试在 Web 上查找实体前, 系统可以使用 URN 先在本地查找。它还允许 Web 位置发生改变, 而仍然能够找到实体。
- use case (使用案例)** (n.)一项特定的最终用户任务或一组任务, 由 [distributed enterprise application \(分布式企业应用程序 \)](#) 执行, 并被用来作为设计、测试和衡量应用程序性能的基础。
- user (用户)** (1) (n.)使用应用程序的人员或服务。从程序学角度而言, 用户由用户名、密码以及应用程序借以识别用户的属性集组成。
- (2) (n.) 经过验证的单个 (或应用程序) 实体。用户可以拥有一组与该身份相关联的角色, 使用户有权访问由这些角色保护的所有资源。另参见 [principal \(主体 \)](#)、[group \(组 \)](#) 以及 [role \(角色 \)](#)。
- user account (用户帐户)** (n.)访问服务器时所用的帐户, 它在目录服务器上以条目形式维护。
- userAgent** (n.)对于 Portal Server Mobile Access, 它是引用 HTTP 用户代理标头的属性。用户代理消息头对于特定移动设备通常是唯一的, 并且可用来检测和检索某客户机类型的数据。

user agent (用户代理)	(n.)允许用户创建、发送和接收邮件的客户机组件，如 Netscape™ Communicator。又称为 UA。
user data constraint (用户数据约束)	(n.)指出应如何保护客户机和 Web 容器之间的数据。保护措施包括避免篡改数据或避免窃听数据。
user entry (用户条目)	(n.)描述每个用户信息的字段（必填和可选）。例如，标识名、全名、称谓、电话号码、寻呼机号码、登录名、密码、主目录等等。又称为用户配置文件。
user folders (用户文件夹)	(n.)用户的电子邮箱。
user group (用户组)	(n.)Message Queue 客户机用户所属的组，其目的是为了授权访问 Message Queue 消息服务器资源，如连接和目的地。
User/Groups Directory Server (用户/组 Directory Server)	(n.)维护组织中用户及组信息的 Directory Server。
user quota (用户配额)	(n.)分配给用户的电子邮件空间总量，由系统管理员进行配置。
user provisioning (用户置备)	(n.)使服务对最终用户可用或向最终用户提供服务访问权的过程。置备涉及身份认证、策略和用户帐户管理活动，如在目录中为每个最终用户创建帐户并以各种服务所需的用户特定信息填充帐户。
user session (用户会话)	(n.)服务器跟踪的一系列用户应用程序交互。会话维护着用户状态、持久对象以及身份验证。

V

- valid (有效的)** (adj.)有效的 XML 文档除了格式正确外，还遵循由 DTD 施加的所有约束。它不包含任何 DTD 所不允许的标记，并且标记顺序也符合 DTD 的规范。
- validating parser (验证解析程序)** (n.)一种 parser (解析程序)，可确保 XML 文档除格式正确外还具备有效性。
- value-binding expression (值绑定表达式)** (n.)一种 JavaServer Faces expression language (JavaServer Faces 表达式语言) 表达式，它引用支持 bean 的属性。组件标记使用该表达式将相关组件的值或组件实例绑定到 bean 属性上。如果组件标记通过其值属性来引用 property，则组件的值限定于 property。如果组件标记通过其绑定属性来引用 property，则组件本身限定于 property。
- vanity domain (虚域)** (n.)与单个用户关联而不与特定服务器或托管域关联的域名。虚域通过使用 MailAlternateAddress 属性指定。虚域没有与域名相对应的 LDAP 条目。虚域适用于希望自定义域名而不想负担支持自身托管域所需管理开销的个人或小型组织。又称为自定义域。
- /var/mail** (n.)此名称通常用来指 Berkeley 式收件箱，在此类收件箱中，新邮件依序存储在单个平面文本文件中。
- versioning (版本更新)** 参见 dynamic reloading (动态重新加载)。
- virtual data view (虚拟数据视图)** (n.)JDBC 数据源、LDIF 数据源或多个聚合数据源的 LDAP 表示。虚拟数据视图本质上是常规的 Directory Proxy Server 数据视图，其上定义了某些变换操作。
- virtual domain (虚拟域)** (1) (n.)ISP 托管域。
(2) (n.)由 Messaging Multiplexor 添加到客户机用户 ID 的域名，用于 LDAP 搜索和登录到邮箱服务器。另参见 domain (域)
- virtual host (虚拟主机)** (n.)映射到单个 IP 地址的多个主机及域名。
- virtual list view index (虚拟列表视图索引)** (n.)一种过滤方法，如果带有用户界面的客户机使用虚拟列表视图扩展，采用此方法可加快 Directory Server 控制台 (或其他图形用户界面) 中条目的显示速度。可在目录树的任何分支上创建虚拟列表视图索引来提高特定搜索的显示性能。又称为浏览索引。

virtual private network (虚拟专用网络)	(n.)一种网络，它具有常规网络的外观和功能，但与公共网络中的专用网络相似。通过在较低协议层进行加密，可通过未实施这种加密的不安全网络（通常是 Internet）进行安全连接。VPN 通常比使用专线的真正专用网络成本低。VPN 依靠在两端使用相同的加密系统。加密可由防火墙软件或路由器执行。
virtual server (虚拟服务器)	(1) (n.)针对特定 URL 提供内容服务的虚拟 Web 服务器。多个虚拟服务器可以使用相同或不同的主机名、端口号或 IP 地址来提供内容服务。HTTP 服务可根据 URL 将收到的 Web 请求定向到不同的虚拟服务器。又称为虚拟主机。 (2) (n.)通过虚拟服务器可以为一个已安装的服务器设置多个域名、IP 地址以及服务器监视功能。
virtual server class (虚拟服务器类)	(n.)共享 obj.conf 文件中相同基本配置信息的虚拟服务器集合。
virtual transformation (虚拟变换)	(n.)用于确定如何在 Directory Proxy Server 虚拟数据视图中显示物理数据的定义。虚拟变换在数据视图上定义，以获得数据的其他视图。
voice Portal Desktop (语音 Portal 桌面)	(n.)通过电话或类似设备所呈现的 Portal Server 站点的音频表现形式。
voiceXML	(n.)一种标记语言，用于为交互式语音应答应用程序创建音频对话。
VoIP	(IP 语音) (n.)基于 IP 网络提供语音通话的技术。
volume manager (卷管理器)	(n.)一种软件产品，通过磁盘分散读写、级联、镜像以及元设备或卷的动态增长来提供数据可靠性。
VPN	参见 virtual private network (虚拟专用网络) 。
VPN gateway (VPN 网关)	(n.)VPN 的入口点。通常受防火墙保护。
VRFY	(n.)用于检验用户名的 SMTP 命令。在 RFC 821 中定义。

W

- W3C** (World Wide Web Consortium, 万维网联盟) (n.)控制 Internet 标准的国际组织。其 Web 站点为 <http://www.w3.org/>。
- WAP** (Wireless Application Protocol, 无线应用程序协议) (n.)一种通过无线通信运行应用程序的开放标准。
- WAR file (WAR 文件)** 参见 [web application archive \(Web 应用程序归档\)](#)。
- warning (警告)** (n.)如果文档的 DTD 包含重复性定义, 这些定义在类似情况下不足以成为错误, 但文档作者可能想要了解此情况 (因为它可能成为错误), 如此便会生成 SAX 解析程序警告。另参见 [fatal error \(致命错误\)](#)。
- WCAP** (Web Calendar Access Protocol, Web 日历访问协议) (n.)一种基于命令的高级协议, 客户机使用此协议与 Calendar Server 进行通信。
- web application (Web 应用程序)** (n.)由 servlet、采用 [JSP technology \(JSP 技术\)](#) 创建的页面、HTML 文档以及其他 Web 资源组成的集合, 其中可能包括图像文件、压缩归档及其他数据。Web 应用程序可以打包成 Web 归档 (WAR 文件), 也可存在于开放目录结构中。Java Enterprise System Application Server 还支持一些非 Java Web 应用程序技术, 如 [SHTML](#) 和 [CGI](#)。
- web application archive (Web 应用程序归档)** (n.)一种归档文件, 内含压缩形式的完整 Web 应用程序。Java Enterprise System Web Server 不能访问 WAR 文件中的应用程序。必须先解压缩 Web 应用程序 (使用 [wdeploy](#) 实用程序进行部署), Java Enterprise System Web Server 方可为其提供服务。
- web application, distributable (可分布的 Web 应用程序)** (n.)使用 J2EE 技术编写的 Web 应用程序, 这样的应用程序可以在分布于多个 Java 虚拟机 (在相同或不同主机上运行) 上的 Web 容器中部署。此类应用程序的部署描述符使用可分布的元素。
- web cache (Web 高速缓存)** (n.)Java Enterprise System Application Server 的一个功能, 利用此功能, servlet 或采用 [JSP technology \(JSP 技术\)](#) 创建的页面可将其结果在高速缓存中持续存放特定的一段时间, 以提高性能。在这段持续时间内对该 servlet 或 JSP 页的后续调用将得到高速缓存中的结果, 这样, 该 servlet 或 JSP 页便不必重新执行了。
- web component (Web 组件)** (n.)为响应请求而提供服务的一种组件, 可以是 [servlet](#) 或 [JSP page \(JSP 页\)](#)。

web connector plug-in (Web 连接器插件)	(n.)对 Web 服务器进行的扩展, 使得 Web 服务器能够与 Java Enterprise System Application Server 进行通信。
web container (Web 容器)	(n.)实现 J2EE 体系结构的 Web 组件约定的 container (容器) 。该约定指定 Web 组件的运行环境, 包括安全性、并发性、生命周期管理、事务、部署以及其他服务。Web 容器提供与 JSP 容器相同的服务以及 J2EE 平台 API 的联合视图。Web 容器由 Web 或 J2EE 服务器提供。
web container, distributed (可分布的 Web 容器)	(n.)一种 Web 容器, 它可以运行标记为 可分布 并在多个 Java 虚拟机 (在相同或不同主机上运行) 上执行的 Web 应用程序。
web container provider (Web 容器提供者)	(n.)提供 Web 容器的供应商。
webmail (Web 邮件)	(n.)对基于浏览器的电子邮件服务的通称。基于浏览器的客户机称为“瘦”客户机, 因为更多的处理是在服务器上进行的。这种客户机访问的邮件始终存储在服务器上。另参见 Messenger Express 。
web module (Web 模块)	(n.)一种单独部署的 Web 应用程序, 相对于在 J2EE 应用程序中部署。参见 web application (Web 应用程序) 。
web resource (Web 资源)	(n.)包含于可通过 URL 引用的 Web 应用程序中的静态或动态对象。
web resource collection (Web 资源集合)	(n.)一个 URL 模式和 HTTP 方法列表, 用于描述要保护的一组 Web 资源。
Web Server	(n.)Portal Server 中的一种 Web 服务器, 用作 Portal Server 和 Portal Server 包 Web 应用程序的 Web 容器。Sun Java System Web Server 随附在 Directory Server Access Management Edition 产品中。
web server (Web 服务器)	(n) 一种主机, 提供访问 Internet、内联网或外联网的服务, 以及存储和管理 Web 应用程序, 但非所有的 J2EE 应用程序。Web 服务器托管 Web 站点, 提供对 HTTP 和其他协议的支持, 并执行具有特定功能的服务器端程序 (如 CGI 脚本或 servlet)。在 J2EE 体系结构中, Web 服务器向 Web 容器提供服务。例如, Web 容器通常依靠 Web 服务器来提供 HTTP 消息处理。J2EE 体系结构假设 Web 容器由相同供应商提供的 Web 服务器托管, 所以它不指定这两个实体之间的订约。一台 Web 服务器可以容纳一个或多个 Web 容器。
web server plug-in (Web 服务器插件)	(n.)一种 HTTP 反向代理插件, 借此可指示 Java Enterprise System Web Server 或 Java Enterprise System Application Server 将特定 HTTP 请求转发给另一服务器。
web server provider (Web 服务器提供者)	(n.)提供 Web 服务器的供应商。
web service (Web 服务)	(1)(n.)一种服务, 它符合为实现可访问性、服务封装和发现功能而制订的标准化 Internet 协议。这些标准包括 SOAP (简单对象访问协议) 消息传送协议、WSDL (Web 服务定义语言) 接口定义以及 UDDI (通用发现、描述和集成) 注册标准。Web 服务先接受请求, 然后根据请求执行其功能并返回响应。请求和响应可以是同一操作的组成部分, 或者它们也可以单独发生, 在这种情况下使用方无需等待响应。请求和响应通常都采用 XML 的形式 (一种可移植的数据交换格式), 并通过有线通信协议 (如 HTTP) 传送。

	(2)(n.)通过 Web 提供的一种服务。一种自包含、自描述的模块化应用程序，可以跨 Internet 或内联网接受来自系统的请求、对该请求进行处理并返回响应。
web service consumer (Web 服务使用者)	(n.)Web 服务使用者通过向 Web 服务提供者发出请求，调用 Web 服务所提供的操作。
web service provider (Web 服务提供者)	(n.)Web 服务提供者根据 Web 服务使用者发来的请求履行 Web 服务。它可以与使用它的 Web 服务使用者在同一 Java™ 虚拟机上运行。
well-formed (格式正确)	(adj.)语法正确的 XML 文档。它没有任何非标记组成部分的尖括号，所有标记都有结束标记或者其本身就是结束标记，并且所有标记都是完全嵌套的。知道文档格式正确便可对其进行处理。但是，格式正确的文档可能并不有效。要确定有效性，需要 validating parser (验证解析程序) 和 DTD 。
Windows CGI	(n.) (仅限 Windows NT) 采用基于 Windows 的编程语言 (如 Visual Basic) 编写的 CGI 程序。
wireless desktop dispatcher (无线桌面调度程序)	(n.)一个组件，它确定了将用户请求路由到哪个标准 Portal 桌面、移动 Portal 桌面或语音 Portal 桌面。
withdrawn patch (删除的修补程序)	(n.)已从分发系统中删除的修补程序。
WML	(wireless markup language, 无线标记语言) (n.)一种基于 XML 的标记语言，它是 WAP 的组成部分。
workgroup (工作组)	(n.)本地工作组环境，在该环境中，服务器在本地办公室或工作组内执行自身的路由选择和传送。部门间的邮件被路由到主干服务器。另参见 backbone (主干)。
WS-BPEL	(web services business process execution language, Web 服务业务流程执行语言) (n.)BPEL 的一种变体，使用源自 Web 服务描述语言 (web services description language, WSDL) 的构造。WS-BPEL 使用这些构造来描述入站和出站流程接口，这样就可以轻易地将流程集成到其他流程或应用中。
WSDL	(web services description language, Web 服务描述语言) (n.)一种基于 XML 的语言，用于以标准化方式定义 Web 服务。描述 Web 服务的三个基本属性：Web 服务的定义、该 Web 服务的访问方式以及该 Web 服务的位置。

X 到 Z

X.400	(n.)一种消息处理系统标准。
X.500 standard (X.500 标准)	(n.)ISO/ITU-T 文档集，其中概括了建议由 Directory Server 实现使用的信息模型、对象类和属性。 LDAP 是 X.500 标准所使用的“目录访问协议”(Directory Access Protocol, DAP) 的轻量版本。
Xalan	(n.) XSLT 的解释版本。
XA protocol (XA 协议)	(n.)用于分布式事务的数据库行业标准协议。
XHTML	(extensible hypertext markup language , 可扩展超文本标记语言) (n.)HTML 4.0 的改订方案，可通过添加新元素和属性对其进行扩展。与 XML 相似的 HTML 版本，由数个 XHTML DTD 之一所定义。要将 XHTML 用于所有内容当然会有违 XML 的初衷，因为 XML 是要确定信息内容，而不仅仅是指出如何显示它。可以在 DTD 中引用它，这样做有很多优点，例如元素中的文本可以包含 <code></code> 和 <code></code> 标记而不局限于纯文本。
XLink	(n.)XLL 规范的组成部分，与指定文档之间的链接相关。
XLL	(n.)XML 链接语言规范，由 XLink 和 XPointer 组成。
XML	(extensible markup language , 可扩展标记语言) (n.)由万维网联盟 (World Wide Web Consortium, W3C) 开发的一种灵活的编程语言，用于创建公用信息格式以及在 Web、内联网和其他领域实现格式及数据共享。该标记语言允许定义标识 XML 文档中的内容、数据和文本时所需的标记。它不同于 HTML，后者是在 Internet 上显示信息时最常用的标记语言。HTML 具有主要处理样式和显示方面问题的固定标记。XML 文档必须先变换为一种所含样式标记受控于样式表的语言，才能由浏览器或其他显示机制进行显示。与 XML 配合使用的两种样式表类型分别为 CSS 和 XSL 。通常，XML 变换为 HTML 以进行显示。尽管标记可以在生成 XML 文档的过程中根据需要进行定义，但文档类型定义 (document type definition, DTD) 却可以用于定义特定类型文档中所允许的元素。可以使用 DTD 中的规则来对文档进行比较，以确定其有效性以及查找文档中的特定元素。Web 服务应用程序的 J2EE 部署描述符在 XML (具有用于定义所允许元素的模式) 中表示。用于处理 XML 文档的程序使用 SAX 或 DOM API。Calendar Server 使用 XML 和 XSL 生成 Calendar Express 用户界面。

XML namespace (XML 名称空间)	(n.)一种标准, 允许为 DTD (文档类型定义) 定义的元素名称集指定唯一的标签。可将使用该 DTD 的文档加入到其他任何文档中, 而不会引起元素名称间的冲突。这是因为该 DTD 中定义的元素随后会被唯一地标识。举例来说, 如此解析程序便可确定何时应按您的 DTD 而不是另一文档类型定义的 DTD 来解释元素。
XML registry (XML 注册表)	参见 registry (注册表) 。
XML schema (XML 模式)	(n.)用于定义 XML 文档结构、内容和语义的 W3C 规范。
XPath	(n.)用于标识 XML 文档组成部分的寻址机制。
XPointer	(n.)XLL 规范中与标识文档各部分相关的组成部分, 对文档进行标识的目的是在链接中引用这些文档部分或将其包括在其他文档中。
XSL	(extensible style language , 可扩展样式语言) (n.)一种用于创建 XML 样式表的语言, 它与用于 HTML 的层叠样式表 (cascading style sheet, CSS) 类似。在 XML 中, 内容和表示是分开的。XML 标记并不指出其显示方式。XML 文档必须进行格式化才可读取。XSL 标准能让您完成以下操作: <ul style="list-style-type: none">■ 指定寻址机制, 这样就可以标识被变换的文档的各部分 (XPath)。■ 指定标记转换, 这样就可以将 XML 数据转换为不同格式 (XSLT)。■ 指定显示特性, 如每个页面的页面大小、页边距和字体高度与宽度以及流对象。信息先填充在页面的一个区域中, 然后在该区域填满后自动流向下一个对象。这样便可用来环绕图像, 或在另一不同页面上继续完成通讯文章 (XSL-FO)。
XSL-FO	(n.)XSL 的一种子组件, 用于描述字体大小、页面版式以及信息如何从一个页面流向另一个页面。
XSLT	(extensible style language transformation , 可扩展样式语言变换) (n.)XML 样式表在将一种 XML 文档形式转变为另一种 XML 形式时所使用的语言。这种转变在电子商务和电子业务中特别有用, 因为它可为多个平台上的各种 XML 文档编码提供一个通用的标准。目标文档通常具有与显示相关的标记, 用于规定浏览器或其他显示机制对文档进行转译的方式。XSLT 以前是 XSL 的一部分, 后者也包括样式流对象的标记语言。
XSLTC	(n.)XSLT 的编译版本。
Zulu time (祖鲁时间)	(n.)GMT 和 UTC (coordinated universal time , 协调通用时间) 的军方称呼。